APL2

Installation and Customization under CMS

Version 2 Release 2

SH21-1062-01

APL2

Installation and Customization under CMS

Version 2 Release 2

SH21-1062-01

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page vi.

Second Edition (March 1994)

This edition replaces and makes obsolete the previous edition, SH21-1062-0. The technical changes for this edition are summarized under “Summary of Changes,” and are indicated by a vertical bar to the left of a change.

This edition applies to Version 2 Release 2 of APL2, 5688-228, Release 2 of APL2 Version 2 Application Environment, Program Number 5688-229, and to any subsequent releases until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for reader's comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, Department J58
P.O. Box 49023
San Jose, CA, 95161-9023
United States of America

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1984, 1994. All rights reserved.**
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices vi
Programming Interface Information vi
Trademarks vii

About This Book viii
Who Should Read This Book viii
APL2 Publications viii
Conventions Used in This Library ix

Summary of Changes x
Product x

Chapter 1. Overview of APL2 and the Installation Process 1
APL2 1
How APL2 Is Packaged in CMS 1
The Installation Process 2

Chapter 2. Planning the Installation of APL2 3
Preplanning Activity 3
System Requirements 3
Software 3
Hardware 4
Defining the Installation User ID and Minidisk 4
Determining Where the APL2 Modules and System Files Are to Reside 5
Determining Where the APL2 Libraries Are to Reside 5
Planning for Printer Support 6
Planning for Installation of the Global SVP 6
Planning for Installation of the APL2 Port Server 7
Planning for APL2 Saved Segments 8
Defining the APL2 Shared Code Saved Segment 8
Defining the APL2 Global Shared Storage Saved Segment 10
Modifying System Options in AP2VIOPT or AP2VIOPE 10
Modifying APL2PI Command Options in AP2XAPIC 10
Determining APL2 Storage Requirements 11
Planning for Migration from VS APL 11
Planning for Migration from Previous APL2 Releases 12
Determining Post-Installation Customizing Requirements 13
Planning for Data Security 13
Planning for GDDM 13

Chapter 3. Installing APL2 14
Installation Summary 14
Installation Steps 14
Step 1–Define Saved Segments and IMAGELIBs (Optional) 14
Step 2–Define the Product User IDs (Optional) 15
Step 3–Load Installation EXEC and MEMO (Required) 15
Step 4–Perform Default or Customized Installation (Required) 16
Step 5–Review APL2 Installation (Required) 21
Step 6–Copy Files to Production Disks (Optional) 21

© Copyright IBM Corp. 1984, 1994

Chapter 4. APL2 Invocation and Termination 25
Starting APL2—The APL2 Command 25
Continuing with the Invocation Options 26
Description of Invocation Options 28
Session Termination 44

Chapter 5. Customizing APL2 46
Customizing IBM-Supplied Sample Files 46
Customizing Your APL2 Installation Process 46
NAMES Files and the AP2M EXECs 46
Outline of Customizing Installation Steps 49
Installing to a Different Disk and/or Retrieving Optional Material 49
Removing GDDM 51
Removing SQL/DS 51
Adding 3800 Font Support 51
Testing APL2 Without Saving Segments 52
Customizing Saved Segment Information for APL2 52
Adding Saved Segment Information for GDDM and SQL/DS 52
Including a Modified APLSCOPT TEXT Deck 53
Making User-Written Auxiliary Processors Part of the System 53
Making User-Written External Routines Part of the System 55
Installing Namespaces and Non-APL Routines in VM Saved Segments . . . 56
Customizing the Execution of APL2 61
Customizing the APL2 Session Manager Profile 61
Changing the APL2 System and Invocation Options 62
Changing the APL2PI Command Options 63
Using Nondefault Saved Segment Names for APL2 63
Modifying the AP2EXIT EXEC 64
Defining APL2 Libraries 64
Installation Exit for the Global SVP 67
Preparing for Access of SQL/DS by APL2 68
Extending or Adding National Language Message and Command Files . . . 70
Extending or Adding National Language Help Files 71
Providing a Local Greeting Message 72

Chapter 6. Administering the APL2 Environment 74
Running the Global SVP Service Machine 74
Running the Port Server 75
Documenting Problems 78
Maintaining APL2 78
Preventive Service with PUT 78
Corrective Service 79

Appendix A. Sample User IDs and Disk Tables 81
Sample Installation User ID 81
Sample APL2 User User ID 81
Sample Global SVP User ID 81
Sample Port Server User ID 82
Disk Tables for Selected CMS Devices 82

Appendix B. Sample Files 83
AP2EXIT EXEC Sample — AP2EXIT AP2MSAMP 83
AP2LOCAL NAMES Sample — AP2LOCAL AP2MSAMP 85
AP2MPSRV EXEC Sample — AP2MPSRV AP2MSAMP 91

Sample Exit for Global SVP — AP2SVPEX AP2MSAMP 92
TCP/IP Profile File — AP2TCPIP APL2PROF 101
AP2VIOPE ASSEMBLE Sample — AP2VIOPE AP2MSAMP 104
AP2VIOPT ASSEMBLE Sample — AP2VIOPT AP2MSAMP 107
AP2XAPIC Module Sample — AP2XAPIC AP2MSAMP 110
DEFAULT VSAPLPR Sample — DEFAULT AP2MSAMP 113
LIBTAB APL2 Sample — LIBTAB AP2MSAMP 114
NAMESYS Macro Sample — NAMESYS AP2MSAMP 116
NAME3800 Macro Sample — NAME3800 AP2MSAMP 118
RDEVICE Macro Sample — RDEVICE AP2MSAMP 119

Appendix C. Sample Installation 120
Sample APL2 Installation 120
Sample Application Environment Installation 125

Appendix D. APL2 System Options 129
System Options 129
Invocation Options 131

Appendix E. APL2 Files and Data Sets 133
Files 133
CMS FILEDEF (DD) Names 134

Appendix F. AP2M220 NAMES File 135

Appendix G. SCRIPT/VS Support 168
Box Character Set 168
Line Printer Font Information 168

Bibliography 169
APL2 Publications 169
Other Books You Might Need 169

Index 171

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject material in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Corporation, IBM Director of Licensing, 208 Harbor Drive, Stamford, Connecticut, United States 06904.

Programming Interface Information

This installation guide is intended to help you install APL2 Version 2 on CMS.

This book also documents Product-sensitive Programming Interface and Associated Guidance Information provided by APL2.

Product-sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of APL2. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Product-sensitive programming interface

Product-sensitive Programming Interface and Associated Guidance Information...

End of Product-sensitive programming interface

© Copyright IBM Corp. 1984, 1994

Trademarks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

AIX/6000	RETAIN
APL2	SQL
APL2/6000	SQL/DS
GDDM	System/370
IBM	System/390
OS/2	VM/ESA
QMF	VM/XA

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies:

Sun	Sun Microsystems, Inc.
Solaris	Sun Microsystems, Inc.

About This Book

This book is intended to assist you in installing IBM* APL2* under CMS.

Who Should Read This Book

This book, *APL2/370 Installation and Customization under CMS*, is for the system programmer or administrator responsible for installing and customizing the APL2 Licensed Program, or APL2 Application Environment, under CMS. It assumes that you are an experienced VM CMS user, are familiar with the VM CMS environment, and have experience with the installation of licensed programs such as APL2.

This book tells you how to install APL2, tailor the system to meet your installation's requirements, and maintain APL2. It summarizes the major features of APL2 and describes the sequence of tasks you should follow to install, customize, and maintain APL2 under CMS.

APL2 Publications

Figure 1 lists the books in the APL2 library. This table shows the books and how they can help you with specific tasks.

Figure 1 (Page 1 of 2). APL2 Publications

Information	Book	Publication Number
General product	*APL2 Fact Sheet*	GH21-1090
Warranty	*APL2/370 Application Environment Licensed Program Specifications*	GH21-1063
	APL2/370 Licensed Program Specifications	GH21-1070
	APL2 for AIX/6000 Licensed Program Specifications	GC23-3058
	APL2 for Sun Solaris Licensed Program Specifications	GC26-3359
Introductory language material	*APL2 Programming: An Introduction to APL2*	SH21-1073
Common reference material	*APL2 Programming: Language Reference*	SH21-1061
	APL2 Reference Summary	SX26-3999
System interface	*APL2/370 Programming: System Services Reference*	SH21-1056
	APL2/370 Programming: Using the Supplied Routines	SH21-1054
	APL2/370 Programming: Processor Interface Reference	SH21-1058
	APL2 for OS/2: User's Guide	SH21-1091
	APL2 for Sun Solaris: User's Guide	SH21-1092
	APL2 for AIX/6000: User's Guide	SC23-3051
	APL2 GRAPHPAK: User's Guide and Reference	SH21-1074
	APL2 Programming: Using Structured Query Language	SH21-1057
	APL2 Migration Guide	SH21-1069

© Copyright IBM Corp. 1984, 1994

Figure 1 (Page 2 of 2). APL2 Publications

Information	Book	Publication Number
Mainframe system programming	*APL2/370 Installation and Customization under CMS*	SH21-1062
	APL2/370 Installation and Customization under TSO	SH21-1055
	APL2/370 Messages and Codes	SH21-1059
	APL2/370 Diagnosis Guide	LY27-9601

For the titles and order numbers of other related publications, see the "Bibliography" on page 169.

Conventions Used in This Library

This section discusses the conventions used in this library.

lower Lowercase italicized words in syntax represent values you must provide.

UPPER In syntax blocks, uppercase words in an APL character set represent keywords that you must enter exactly as shown.

[ ] Usually, brackets are used to delimit optional portions of syntax; however, where APL2 function editor commands or fragments of code are shown, brackets are part of the syntax.

[*A* | *B* | *C***]** A list of options separated by | and enclosed in brackets indicates that you can select one of the listed options. Here, for example, you could specify either *A*, *B*, *C*, or none of the options.

{*A* | *B* | *C***}** Braces enclose a list of options (separated by |), one of which you must select. Here, for example, you would specify either *A*, *B*, or *C*.

... An ellipsis indicates that the preceding syntactic item can be repeated.

{}... An ellipsis following syntax that is enclosed in braces indicates that the enclosed syntactic item can be repeated.

The term *workstation* refers to all platforms where APL2 is implemented except those based on System/370* and System/390* architecture.

Throughout this book, the following product names apply:

Product Name	Platform
APL2/2	OS/2*
APL2 for Sun Solaris	Sun** Solaris**
APL2/6000*	AIX/6000*
APL2/370	MVS or VM
APL2/PC	DOS

Summary of Changes

Product

APL2/370, Version 2 Release 2

Date of Publication: March 1994

Form of Publication: Revision, SH21-1062-01

Document Changes

- Updated all references to version and release levels.
- Added planning information for GDDM*.
- Added NLT invocation option.

© Copyright IBM Corp. 1984, 1994

Chapter 1. Overview of APL2 and the Installation Process

This chapter summarizes the major features and components of the APL2 Licensed Program and provides an overview of the installation process.

APL2

APL is a general purpose programming language used in a variety of applications such as commercial data processing, systems design, prototyping, and scientific and engineering computations. APL2 is an implementation of APL that generalizes many of its fundamental concepts, extends many APL functions and operators, and removes many restrictions present in other implementations of APL.

The APL2 Licensed Program consists of a language processor and supporting services. The language processor includes operations for manipulating APL2 data and data structures. The supporting services allow for interaction with the host system, data management, and error handling.

Most APL2 installations support APL2 programming, as well as personal computing and APL2 applications. A careful selection of the options for installing and customizing APL2 allows you to accommodate the requirements of various user groups.

APL2 Application Environment is a subset of the APL2 Licensed Program that provides the APL2 environment necessary to run prewritten APL2 application packages.

APL2 Application Environment:

- Provides the ability to run applications that have been packaged using the full APL2 product.
- Can be used as an alternative to APL2 in those installations or systems where development facilities are not required.

The following facilities, provided with APL2, are **not** available with APL2 Application Environment:

- The session manager
- The editors
- The library workspaces
- Certain system commands
- Certain external routines
- Shared variable interpreter interface

Information that does not apply to APL2 Application Environment is noted within the text.

How APL2 Is Packaged in CMS

APL2 executes as an ordinary CMS command. An invocation module locates the main code as a nucleus extension and then transfers control to the main code for the duration of the APL session. The bulk of APL2 code is designed to reside in several saved segments, which may be shared among all APL users. If any of the saved segments are unavailable, either because they are not defined or would not

© Copyright IBM Corp. 1984, 1994

load in virtual machine, an unshared copy is fetched from one of several LOADLIBs.

During the session, APL2 programs can retrieve, manipulate, and update information from the CMS system through a set of special interfaces that provide access to the file system, program stack, commands, and other products such as the Graphical Data Display Manager (GDDM) and Structured Query Language/Data System (SQL/DS*). APL2 itself uses standard CMS interfaces wherever possible, but it must still rely on OS simulation for some services, such as ABEND recovery (STAE), timer services (STIMER) and attention handling (STAX).

The global SVP, an optional feature, lets users transfer data between virtual machines using an unprotected saved segment and the Virtual Machine Communication Facility (VMCF) identify protocol.

The Installation Process

To install APL2, you must do three things:

- **Plan the Installation**

  You should complete a thorough installation plan before installing APL2. This involves ensuring that system and storage requirements have been met, deciding how APL2 is installed, and completing all preinstallation tasks. See Chapter 2, “Planning the Installation of APL2” on page 3 for information on developing your plan.

- **Install the APL2 Licensed Program**

  EXECs from the distribution tape handle the installation of APL2. Instructions for both required and optional installation steps are provided for you in Chapter 3, “Installing APL2” on page 14.

- **Customize APL2**

  In order for you to install APL2 successfully, some aspects of its operation may be tailored, either during or after installation, to meet user requirements. Other aspects can be tailored as needed. Chapter 5, “Customizing APL2” on page 46 discusses the options in detail.

After installation, VS APL users may need to migrate individual VS APL workspaces and files to APL2. This process is discussed in *APL2 Migration Guide*.

You have to provide some ongoing support after APL2 has been installed. System administration includes such tasks as running the Global SVP Service Machine, documenting problems, applying preventive service with PUT, applying corrective service with Program Temporary Fixes (PTFs), adding user-written auxiliary processors or external routines, and providing a local greeting message. Chapter 6, “Administering the APL2 Environment” on page 74 provides you with information on administering APL2 to ensure efficient operation.

Chapter 2. Planning the Installation of APL2

You should complete a thorough installation plan prior to the installation date. The plan helps you to ensure a successful installation of APL2. This chapter discusses the information you need to begin planning and then outlines the system requirements and the decisions to be made about APL2 libraries, system options, auxiliary processors, virtual and auxiliary storage, migrating from VS APL, data integrity, and customizing the system.

Preplanning Activity

This installation guide is supplemented by *APL2 Program Directory*, which comes with your APL2 installation tape from the IBM Software Distribution (SD). The program directory contains information related to installation activities such as:

- APL2 Licensed Program Preventive Service Planning (PSP) Upgrade identifier
- Organization and content of the distribution material
- Any required updates to this guide
- A list of any prerequisite Program Temporary Fixes (PTFs)

Before you begin your planning effort, it is important that you check the program directory for updates to the information and procedures in this guide and the RETAIN*/370 Preventive Service Planning (PSP) Facility for a list of prerequisite PTFs. To order the prerequisite PTFs contact your IBM Support Center.

System Requirements

APL2 runs on the software and hardware products listed below. For more information and updates see *APL2 Program Directory*.

Software

Under CMS, APL2 runs on the VM/System Product Licensed Program (VM/SP), Release 6 or later, VM/XA* System Product, Release 2.1 or later, or on VM/ESA* System Product, Release 1.1 or later.

Use of the DSQCIA External Function requires QMF* Version 3 Release 1 or later.

Use of AP 119 or cross system shared variables requires TCP/IP Version 2 Release 1 or later.

Use of AP 127, the Structured Query Language (SQL*) auxiliary processor, requires the IBM SQL/DS Licensed Program, Version 3 Release 3 or later.

Use of AP 123 requires VSAM.

Use of the following APL2 features requires the Graphical Data Display Manager (GDDM), Version 2 Release 3 or later:

- APL2 session manager
- APL2 full-screen editor—Editor 2
- APL2 Session Manager Command auxiliary processor—AP 120

© Copyright IBM Corp. 1984, 1994

- GDDM auxiliary processor—AP 126
- Any of the following supplied workspaces: *GRAPHPAK*, *FSM*, *FSC*126, *GDMX*, and *CHARTX*

Of the features requiring GDDM, only the GDDM auxiliary processor, AP 126, is included with APL2 Application Environment.

Hardware

System

APL2 is designed to run on any IBM System/370 or System/390 processor that also runs a supported release of VM.

Terminals

APL2 under CMS is supported on the IBM 3270 Information Display System family of terminals with appropriate APL features, as well as terminals and terminal emulators that are functionally equivalent. IBM assumes no responsibility for establishing functional equivalency.

APL2 can run on the 3270 display terminal even if it does not have APL features. However, entering and displaying certain APL characters on the terminal requires the APL features listed above. IBM recommends that each installation have at least one terminal equipped with APL features.

If an APL terminal is not available:

- Users are not able to enter or display the APL special characters
- Editing and correcting of APL2 functions is difficult and awkward; functions in the MEDIT workspace are provided to facilitate this task
- The text of APL2 error messages does not display APL special characters
- The full range of APL2 capabilities are not available

Control Units

In order for the APL characters to be available, terminals with APL features, particularly 3270 family terminals, must be connected to control units that support the APL character set. You should verify that APL features are available and that the proper customizing for APL use has been performed on your installation's control units.

Auxiliary Storage

APL2 uses standard system interfaces to access auxiliary storage. It supports any auxiliary storage devices that are supported by those interfaces.

Defining the Installation User ID and Minidisk

IBM recommends that you define a unique user ID for the installation of the APL2 Licensed Program. This practice avoids conflict with any prior installation of other APL products.

If you have a prior release of the APL2 Licensed Program installed, you should assign a different minidisk for the installation of this release of APL2. Some parts

of APL2 may not be compatible from release to release, and mixing parts from several releases on one minidisk may cause errors.

By default, the APL2 installation EXEC installs APL2 on your A-disk. If you do not define a unique user ID, you should assign a unique minidisk for the APL2 Licensed Program, and access this disk as your A-disk while you are installing APL2.

The user ID for the installation must have:

- Privilege class E if APL2 code is to be saved into saved segments, and privilege class A, B, or C if IBM 3800 line-mode fonts are to be installed.
- A maximum storage larger than the largest saved segment end address if you are going to save a shared system, or maximum storage of at least 4 megabytes if you are not.

Space requirements for the A-disk are shown in Figure 7 on page 82.

Determining Where the APL2 Modules and System Files Are to Reside

During installation, files are copied from the installation tape to your A-disk. The A-disk is also used for customizing and maintaining APL2 as well as for preproduction testing. For actual production use, you may choose to copy the subset of files required for execution to another disk so that maintenance and testing can be performed on your A-disk without interfering with production use of APL2. During the planning phase, identify the disk to which you optionally copy the APL2 modules and system files. We recommend that this system disk be protected from general write access. See “Planning for Data Security” on page 13 for more suggestions on how to provide data security.

Determining Where the APL2 Libraries Are to Reside

Note: APL2 Application Environment uses libraries only for APL data files. Workspaces are not distributed with APL2 Application Environment.

During installation, the workspaces distributed with APL2 are loaded on your A-disk, with file types consistent for public libraries 1 and 2. The sample LIBTAB APL2 distributed with APL2 points to the A-disk for public libraries 1 through 99. Approximately 2,000 of the 13,600 1K-byte blocks mentioned Figure 7 on page 82 for basic material are required for the workspaces distributed with APL2.

In the final installation step for APL2, you copy the workspaces from your A-disk to a production disk. APL public libraries have traditionally been on the Y-disk, but that is not a requirement. During the planning phase, identify the disk to which you are copying the APL2 public libraries. You have to modify the sample LIBTAB APL2 provided to point to the disk you have selected. See “Defining APL2 Libraries” on page 64 for more information on setting up your libraries.

In addition to the common code in public libraries, each APL2 user normally stores work in a private library. Usually, you want this library to be your A-disk, and the sample distributed by IBM does this. Space requirements for these workspaces depend on the individual user.

Planning for Printer Support

APL programmers and many APL users need to be able to print programs, session logs, and reports containing special APL characters. You may also need to print diagnostic information containing APL characters if you have to report a problem in the product. For these reasons you should plan on having APL fonts available on some or all of your printers. Fonts can be ordered as a hardware feature for impact printers, or can be downloaded from VM to printers in the 3800 family, including 3812, 3820, 3827, etc.

A number of downloadable fonts are included with the APL2 product. They fall into two classes:

- Line-mode fonts for 3800 printers (but not 3812, 3820,etc.). These are 10-pitch and 12-pitch monospaced fonts emulating the behavior of impact printers.
- APA-mode (All Points Addressable) fonts for all 3800 family printers except 3800 model 1. The APL fonts provided are also monospaced, but come in a wide range of sizes and can be intermixed with proportional fonts and graphics.

The line-mode fonts are now distributed largely for compatibility with earlier APL systems. Many installations find that they no longer need them. The APA-mode fonts are required for most remote printers, are fully compatible with IBM Script (DCF) and related products, and are recommended for most installations.

The installation procedures are different for the two classes of fonts, so you need to watch for notations as which procedures apply to “line-mode fonts” and which to “APA fonts.”

Planning for Installation of the Global SVP

The global Shared Variable Processor (SVP) is an optional APL2 facility that runs as a CMS service machine. With the global SVP installed and operative, you can share variables with other APL2 users through an unprotected saved segment.

The Virtual Machine Communication Facility (VMCF) identify protocol is used for communication synchronization between the service machine and APL2 users.

Global SVP usage is permitted either by default, when you invoke APL2 with the ID parameter, or through a user-written installation exit running in the SVP service machine. “Installation Exit for the Global SVP” on page 67 discusses how you may provide an installation exit.

No security is possible for data passed through the global SVP, other than the name of the unprotected saved segment. After you learn the name of the saved segment, it is possible that any part of it might be inspected or modified.

If you plan to install the global SVP, you must:

1. Allocate saved segments for global shared storage. This is described in “Defining the APL2 Global Shared Storage Saved Segment” on page 10.

   “NAMESYS Macro Sample — NAMESYS AP2MSAMP” on page 116 contains a sample saved segment definition.

2. Define the CMS user ID AP2SVP. This user ID is used as the SVP service machine.

   The AP2SVP service machine size is 2M bytes, with an A-disk of 100 1K-byte blocks free. A sample definition of this service machine is shown in "Sample Port Server User ID" on page 82. We recommend that the A-disk for this service machine be protected from read and write access. For more recommendations on providing security, see "Planning for Data Security" on page 13.

   If you want to change the service machine user ID from AP2SVP, then you must plan to modify the APL2 system options parameter CSVPID= in the AP2VIOPT ASSEMBLE or AP2VIOPE ASSEMBLE module during installation. See "Changing the APL2 System and Invocation Options" on page 62 for instructions on how to modify the AP2VIOPT ASSEMBLE or AP2VIOPE ASSEMBLE module. (AP2VIOPT is used with APL2. AP2VIOPE is used with APL2 Application Environment.)

3. Implement an installation exit command for the SVP service machine if you want to assign default identifications to users who communicate using the global SVP. See "Installation Exit for the Global SVP" on page 67 for details on the AP2SVPEX command.

Planning for Installation of the APL2 Port Server

APL2's cross system shared variable facility provides communication across TCP/IP networks through APL2 shared variables. The APL2 port server is an integral part of the cross systems shared variable support.

The APL2 port server is an optional APL2 facility that runs as a separate APL2 session normally in a disconnected machine. Either the APL2 or APL2 Application Environment product can be used to run the server. Although it is possible for users to use cross systems shared variables without a port server, the port server makes the initialization of share variable connections less cumbersome.

If you plan to install the port server, you must:

1. Ensure TCP/IP is installed.
2. Define the CMS user ID APL2PSRV. This user ID is used as the port server service machine. A sample definition of this service machine is shown in Appendix A, "Sample User IDs and Disk Tables."
3. Decide whether the port server should use the default port number.

   By default, the APL2 port server uses TCP/IP port number 31415. If this port number is unavailable on the system, the port server should be started using a different port number. If a port number other than the default is used, you may elect to modify the default APNAMES invocation option value.

4. Determine the default shared variable memory size users should receive.

   Unlike variables shared using the global shared variable processor, cross systems variables are stored in the users' local shared memory. If users require the ability to transfer larger objects than fit in the default shared memory size, you may elect to increase the default SHRSIZE invocation option value to provide larger shared memories.

Planning for APL2 Saved Segments

We recommend that you install the APL2 code as a shared system. A shared system resides in saved segments outside the normal addressing range of your virtual machine. A shared system makes more efficient use of real storage, because multiple users share a single copy of the APL2 code. When one or more APL2 sessions are active, it reduces the load on the system for each additional invocation of APL2 by avoiding the disk and paging activity associated with fetching a new nonshared copy. A shared system usually provides better performance.

On VM/SP, planning for a saved segment requires that you:

- Define the saved segment by coding a NAMESYS entry in the CP module DMKSNT
- Reserve permanent space on a CP-owned volume to hold the saved system image
- Regenerate the CP nucleus to include the new saved segment definition

"NAMESYS Macro Sample — NAMESYS AP2MSAMP" on page 116 contains sample NAMESYS entries that you can use in the module DMKSNT to define the saved segments for APL2 shared code and global shared storage.

On VM/XA SP and some VM/ESA systems, you define saved segments dynamically, using the DEFSEG command. "NAMESYS Macro Sample — NAMESYS AP2MSAMP" on page 116 also contains the equivalent DEFSEG commands.

Other VM/ESA systems support OVERRIDE source files as a means of defining segments, using the DEFSEG tag. DEFSEG tag parameters are equivalent to the parameters used on NAMESYS macros. Substitute values such as shown with NAMESYS AP2MSAMP when using DEFSEG tags in OVERRIDE files.

Defining the APL2 Shared Code Saved Segment

- Beginning with APL2 Version 2 Release 1, the majority of APL2 code for the full APL2 product has been split into two LOADLIBs, APL2 LOADLIB and AP2HIGH LOADLIB. Either LOADLIB may be saved in a saved segment to reduce overall storage requirements and improve performance.
- The majority of code for the APL2 Application Environment has been split into APL2AE LOADLIB and AP2HIAE LOADLIB, which may also be saved in saved segments.
- The saved segment for APL2 LOADLIB or APL2AE LOADLIB must reside below the 16-megabyte line. The saved segment for AP2HIGH LOADLIB or AP2HIAE LOADLIB can reside above the 16-megabyte line when users are running with SET MACHINE XA.
- Multiple instances of each saved segment may be defined, with different names and at different addresses. You might want to save AP2HIGH LOADLIB in a segment above the 16-megabyte line for users running with SET MACHINE XA, for example, and save it again below the 16-megabyte line for users running with SET MACHINE 370.
- For the full APL2 product the segment names you define must match those defined in the installation options module AP2VIOPT. AP2VIOPT as shipped by IBM specifies two names for the APL2 LOADLIB, AP2V22V1 and

AP2V22V2, and two names for the AP2HIGH LOADLIB, AP2V22W1 and AP2V22W2. Sample saved segment definitions can be found in the "NAMESYS Macro Sample — NAMESYS AP2MSAMP" on page 116.

- For the APL2 Application Environment the segment names you define must match those defined in the installation options module AP2VIOPE. AP2VIOPE as shipped by IBM specifies two names for the APL2AE LOADLIB, AP2V22E1 and AP2V22E2, and two names for the AP2HIAE LOADLIB, AP2V22F1 and AP2V22F2. Sample saved segment definitions can be found in the "NAMESYS Macro Sample — NAMESYS AP2MSAMP" on page 116.
- If you choose other names, or if you want more than two copies of one of the segments, you'll need to customize AP2VIOPT or AP2VIOPE during the installation of APL2. You don't need to customize AP2VIOPT or AP2VIOPE if you use one or both of the default names, or if you don't intend to install APL2 in saved segments.
- If you do change the segment names in AP2VIOPT or AP2VIOPE, you must also identify the new names in your AP2LOCAL NAMES file. See "Using Nondefault Saved Segment Names for APL2" on page 63.
- To determine a load address for a saved segment follow these guidelines:
  - On VM/SP, the address must be greater than the largest virtual machine of any APL2 user (or category of user if multiple copies of a saved segment are defined).
  - On VM/XA and VM/ESA, if a saved segment is defined within the address range of a virtual machine, you may have to issue SEGMENT RESERVE for the segment name early in your PROFILE EXEC in order to ensure that it can be used.
  - The address should not be unnecessarily high. If it is, real storage is wasted for unreferenced CP segment table entries.
  - The segment must not overlap other saved segments that are used concurrently, such as those for QMF, GDDM, DCF, VSAM, SQL/DS, ISPF, or the APL global SVP.
- When an individual user invokes the APL2 MODULE, APL2 attempts to use the saved segments in the order in which they appear in AP2VIOPT, and uses the first ones it can successfully load. If APL2 cannot load one successfully it then attempts to load APL2 into the user's virtual machine from the various LOADLIBs.
- The amount of virtual storage required for the APL2 saved segments depends on the size of the LOADLIBs. One megabyte is sufficient for AP2HIGH or AP2HIAE LOADLIB as shipped by IBM. 320 kilobytes is sufficient for APL2, and 256 kilobytes is sufficient for APL2AE LOADLIB. If you make user-written auxiliary processors or namespaces resident with APL2, or customize or service APL2 in any way that could affect the size of the LOADLIBs, you should check the corresponding LKEDIT file to see that the size of the saved segment is still sufficient.

Defining the APL2 Global Shared Storage Saved Segment

The "NAMESYS Macro Sample — NAMESYS AP2MSAMP" on page 116 contains an entry labelled AP2SM22 for an APL2 global shared storage saved segment. Only one global shared storage saved segment should be defined. The address of the global shared storage saved segment is assigned according to the same considerations listed above for the shared code saved segments.

You can choose any name for the global shared storage saved segment. The same name has to be used in the command that starts the global shared variable processor. The sample that IBM supplies uses AP2SM22.

The size of the global shared storage saved segment affects the amount of data that can be shared. The size of the sample saved segment, 0.5 megabytes, is a good choice for an installation with no previous experience using the global SVP. You can adjust the size based on real-time use of the global SVP on your system.

Modifying System Options in AP2VIOPT or AP2VIOPE

APL2 system options are parameters that tailor aspects of the APL2 Licensed Program to the individual requirements of your user community. During the installation of APL2, you may not need to change the values that come with the distributed system. The installation procedures that IBM supplies assume that no values have been modified.

As you gain experience with APL2, there may be some advantage to altering the values to fit your environment. The APL2 system options allow you to make user-written auxiliary processors available by default and specify default and overriding invocation options. VS APL installations that have tailored the system options may want to select the same system options in the APL2 installation.

You can change the default system options during APL2 installation or at any time after installation. To identify system options that may need to be changed, see "Changing the APL2 System and Invocation Options" on page 62 and the default system options summarized in Appendix D, "APL2 System Options" on page 129. Also, consult with the APL2 users or the APL2 administrator to identify the appropriate values for any options to be changed during installation.

Modifying APL2PI Command Options in AP2XAPIC

The APL2PI command options are parameters that tailor the invocation of the APL2 Licensed Program to the individual requirements of the APL2PI user community. During the installation of APL2, you may not need to change the values that come with the distributed system. The APL2PI command options allow you to:

- Specify the name of the APL2 module that should be used when invoking APL2 from APL2PI.
- Specify that the APL2 Application Environment module should be used when invoking APL2 from APL2PI.
- Specify the invocation options that should be used by APL2PI.

The default APL2 invocation module name used in the AP2XAPIC command options module refers to the default name for the full APL2 product invocation module. If this name has been changed, or if the APL2 Application Environment is

to be used, the name in AP2XAPIC should be changed. To identify the appropriate values for any invocation options to be changed during installation, you should consult the APL2PI users or the APL2 administrator.

You can change the default AP2XAPIC command options during APL2 installation or any time after installation. To change the APL2PI command options, see "Changing the APL2PI Command Options" on page 63.

Determining APL2 Storage Requirements

Each APL2 user requires at least 0.5 megabytes of virtual storage for user data.

APL2 code requires approximately 1.33 megabytes of virtual storage. APL2 Application Environment code requires approximately 1.0 megabytes of virtual storage. This number must be multiplied by the number of users who are not using a saved segment, plus the number of saved segments in use.

The sum of the user requirements plus the APL2 code requirements is the virtual storage requirement for APL2 on the host system.

Use of licensed programs such as GDDM and VSAM requires additional storage.

Approximately 4 megabytes of auxiliary storage are required for the system and workspaces provided with the APL2 Licensed Program. Each user's stored workspaces requires additional storage.

Storage requirements also depend on what auxiliary processors are installed and whether the default invocation options, or other APL2 system options, are changed during installation.

If you customize the options to remove APL2 support code for optional licensed programs, you can recover approximately 80K bytes of virtual storage when removing GDDM support and 50K bytes when removing SQL/DS support.

On XA mode virtual machines, APL2 can exploit the storage available above the 16-megabyte line by putting the following above the line:

- The APL2 workspace
- Local shared memory
- Global shared memory
- A saved segment containing much of APL2's own code

Planning for Migration from VS APL

VS APL users must migrate VS APL workspaces to APL2. The APL2 system command)*MCOPY* simplifies this procedure, but additional modification of workspaces may be required. VS APL data files may also require migration. Workspaces and files must be migrated individually. For instructions on how to migrate, see *APL2 Migration Guide*. VS APL users cannot migrate workspaces directly to APL2 Application Environment. VS APL workspaces can only be used with APL2 Application Environment after they have been migrated and packaged using APL2.

Note: Any application that uses APL features not available with APL2 Application Environment, such as)*LOAD*,)*COPY*,)*SAVE*, or terminal I/O other than

through auxiliary processors, may have to be redesigned and rewritten to run under APL2 Application Environment.

The APL2 versions of the auxiliary processors that IBM supplies must be used. VS APL auxiliary processors of the same names and similar functions may not be used with APL2. VS APL user-written auxiliary processors must be reinstalled under APL2 to customize the system. This is true whether or not the processor was resident under VS APL. Adding VS APL auxiliary processors to APL2 is described in "Making User-Written Auxiliary Processors Part of the System" on page 53.

If a VS APL user-written auxiliary processor tries to reference APL2 data that cannot be represented in a VS APL format, it receives a previously undefined return code. Some recoding of the auxiliary processor may be required, depending on how the auxiliary processor reacts to this new return code. For additional information, see *APL2 Migration Guide*.

Planning for Migration from Previous APL2 Releases

Workspaces from the previous version of APL2 do not require migration. They can be directly loaded or copied (using the)*LOAD* command or)*COPY* command, respectively). However, workspaces saved under APL2 Version 2 Release 1 (using the)*SAVE* command) cannot be loaded by any previous APL2 release. Instead, to move a workspace back to any previous APL2 release, you should write it out using the)*OUT* command and then read it in using the)*IN* command under that previous APL2 release.

If you are migrating from Version 1 of APL2, VS APL user-written auxiliary processors must be reinstalled. Even if the EXEC AP2MV1AP had been used to prepare the AP text decks for use under Version 1, the EXEC AP2MV1AP must be used again to prepare the text decks for use under APL2 Version 2. The LOADLIBs produced by Version 1 of AP2MV1AP are not compatible with APL2 Version 2.

| The APL2 Version 2 Release 2 global Shared Variable Processor (SVP) is upward
| compatible from any previous APL2 release. While the APL2 Version 2 Release 2
| global SVP saved segment is active, users of APL2 Version 1 and Version 2
| Release 1 can share variables through it. If only the earlier APL2 release global SVP is active, users of APL2 Version 2 Release 2 are not able to share variables through it.

The global SVP user ID may have only one release of the global SVP active at a
| time. If the Version 2 Release 2 global SVP is to be tested while the earlier
| release global SVP is active, an alternative user ID must be used for Version 2
| Release 2. You must identify the user ID in the CSVPID parameter in the AP2VIOPT ASSEMBLE or AP2VIOPE ASSEMBLE module during installation.

Users of previous releases of the full APL2 product cannot migrate workspaces directly to APL2 Application Environment. Workspaces must be migrated and packaged using APL2 before they can be run under APL2 Application Environment. Workspaces must be redesigned and rewritten if they use features that are not available in APL2 Application Environment.

Determining Post-Installation Customizing Requirements

After installation, you can customize APL2 to meet the particular requirements of its users. During the planning phase, you should consult your APL2 users to determine their specific needs. See Chapter 5, “Customizing APL2” on page 46 for information on how to tailor APL2.

Planning for Data Security

To provide for data security in your APL2 system, consider the following recommendations:

- Minidisks on which private libraries reside should be protected from read and write access.
- Minidisks on which public libraries reside should be protected from write and possibly read access.
- The APL2 installation A-disk should be protected from write access.
- The APL2 system disk should be protected from write access.
- The A-disk for the global SVP service machine, if installed, should be protected from read and write access.

Planning for GDDM

The APL2 session manager assumes GDDM is installed with EBCDIC code page 351 as the default code page. If your installation has customized GDDM so that some other code page is the default, then APL characters may not be handled properly. To correct this, reset the default with a GDDM external defaults file. GDDM external defaults files are described in detail in *GDDM Base Programming Reference Volume 2*, SC33-0332.

Under CMS, you do this by creating a file on an accessed disk. By default, GDDM expects the name for this file to be PROFILE ADMDEFS, but this name can be customized at your installation.

The file or data set should contain the following record:

```
 ADMMDFT APPCPG=351
```

Note that the leading blank is required before ADMMDFT.

Chapter 3. Installing APL2

This chapter contains a summary of the installation procedure as well as step-by-step instructions. Your installation plan contains information that helps you install APL2.

Installation Summary

Under CMS, installation of APL2 is performed by the I5688228 EXEC. This EXEC is contained on the APL2 distribution tape. The tape also contains the load libraries, samples, parameter files, macros, workspaces, symbol sets, and 3800 fonts distributed with APL2.

Installation of APL2 Application Environment is performed using the I5688229 EXEC. This EXEC is contained on the APL2 distribution tape. The tape contains the load libraries, samples, parameter files, macros, and 3800 fonts distributed with APL2. The tape for the I5688229 EXEC does not include workspaces or symbol sets.

Several supplemental steps are required to load the I5688228 or I5688229 EXEC, make any necessary changes to the system options, define APL2 as a shared system (if you are installing a shared system), and verify successful installation. You may need to perform one or more optional installation steps, depending on your particular installation plan.

Installation Steps

This section contains the step-by-step instructions you need to install APL2. Both required and optional steps are described in the order in which they should be performed. If you skip any of the optional steps during the initial installation, you can perform them later during a reinstallation of APL2.

Before beginning the installation, you should review the Program Directory for any updates to these procedures.

Step 1–Define Saved Segments and IMAGELIBs (Optional)

Do this step in preparation for installing:

- APL2 in saved segments
- Global Shared Variable Processor (GSVP)
- 3800 line-mode fonts (not required for 3800/3812/3820 APA fonts)

The specific instructions and commands for defining saved segments and IMAGELIBs vary depending on the level of your VM system (VM/SP, VM/XA, VM/ESA). Consult the Planning and Installation manuals from your level of VM for specific instructions.

1. If you wish to install APL2 in saved segments, define them.

   A discussion of saved segment considerations for APL2 appears under “Planning for APL2 Saved Segments” on page 8. Examples of the NAMESYS macros or DEFSEG commands that may be required to define the APL2 saved

© Copyright IBM Corp. 1984, 1994

segments are shown in “NAMESYS Macro Sample — NAMESYS AP2MSAMP” on page 116.

2. If you are installing the GSVP, define a saved segment for it.

   A discussion of saved segment considerations for APL2 appears under “Planning for APL2 Saved Segments” on page 8. An example of the NAMESYS macro or DEFSEG command that may required to define the GSVP saved segment is shown in “NAMESYS Macro Sample — NAMESYS AP2MSAMP” on page 116.

3. If you are installing the APL line-mode fonts for the 3800 Model 1 printer, define an IMAGELIB for them. See “NAME3800 Macro Sample — NAME3800 AP2MSAMP” on page 118 for information on how to define IMAGELIBs.

Step 2–Define the Product User IDs (Optional)

Do step 2 if you are defining a unique user ID for the APL2 product or if you are installing the global SVP or TCP/IP port server.

1. Define the APL2 Licensed Program installation user ID.

   IBM recommends the user ID APL2. The A-disk allocation is in Appendix A, “Sample User IDs and Disk Tables” on page 81. The virtual machine size must be at least 4 megabytes. If you are building a shared system, the virtual machine must be larger than the largest address in any segment to be saved. The A-disk should be protected from write access by other users. The user ID should have privilege classes E and G if it is used to save a shared segment. The user ID must also have class A, B, or C to install the 3800 fonts. An example of this user ID is shown in Appendix A, “Sample User IDs and Disk Tables” on page 81.

2. If you are installing the global SVP, define the APL2 global SVP user ID. The recommended user ID is AP2SVP. Any user ID can be used, but it must match the value of the system option CSVPID in AP2VIOPT or AP2VIOPE. The system options are described in Appendix D, “APL2 System Options” on page 129.

   The A-disk DASD requirement is 100 1K-byte blocks. An example of this user ID is shown in Appendix A, “Sample User IDs and Disk Tables” on page 81.

3. If you are installing the port server, define the APL2 port server user ID. The recommended user ID is APL2PSRV.

   An example of this user ID is shown in Appendix A, “Sample User IDs and Disk Tables” on page 81.

4. If any of the product user IDs have been defined, load the updated VM directory.

Step 3–Load Installation EXEC and MEMO (Required)

This step loads several files including the APL2 installation EXEC I5688228 or I5688229 from the distribution tape. If you have not performed this step during a previous installation of this release of APL2 on your system, you must perform it now.

1. Log on to the user ID to be used for the APL2 installation.

2. Set the proper virtual machine size — at least 4 megabytes, and larger than any saved segment addresses to be saved:

   CP DEFINE STOR *nn*M

   IPL CMS

3. Spool the console, using the following command:

   CP SPOOL CONSOLE START

   IBM recommends this so that a record of the installation process exists in case of a problem.

4. Access the disk on to which APL2 is to be loaded as your A-disk, unless it is already accessed as such.

5. Mount the distribution tape as virtual device 181.

6. Transfer the first two files from the distribution tape to your A-disk and print the memo to users:

   VMFPLC2 REW

   VMFPLC2 LOAD * * A (EOF 2

   For APL2:

   PRINT I5688228 MEMO (CC

   For APL2 Application Environment:

   PRINT I5688229 MEMO (CC

   For information on the VMFPLC2 command, see the VM library.

   For information about loading and installing distribution tape files to a location other than your A-disk, see “Installing to a Different Disk and/or Retrieving Optional Material” on page 49.

7. Review the printed I5688228 or I5688229 MEMO file for any updated information relating to the installation of APL2.

Step 4–Perform Default or Customized Installation (Required)

Do this step to install APL2. Choose customized installation if you want to alter either the installation process, the APL2 system options, or both.

Customizing the Installation Process

You can alter the APL2 installation process to:

- Install the remaining files to a disk other than your A-disk.
- Include a modified APLSCOPT TEXT deck (for installations that had. VS APL with a modified APLSCOPT TEXT deck)
- Remove APL2 support for GDDM (for installations that do not have, or want, GDDM support).
- Remove APL2 support for SQL/DS (for installations that do not have, or want, SQL/DS support).
- Generate 3800 line-mode fonts (for installations that want, and do not have, 3800 line-mode APL2 fonts).

Customizing the APL2 System Options

You can tailor certain aspects of APL2 operation for your particular system or location. Update the system options when you want to:

- Change the saved segment names used when referencing APL2.
- Change the user ID of the global SVP service machine or port server.
- Make your own auxiliary processors resident with APL2.
- Change default invocation options.

Customizing the APL2PI Command Options

You can tailor how APL2PI invokes APL2 for your system or location. Update the command options when you want to:

- Change the name of the APL2 module that should be used when invoking APL2 from APL2PI.
- Specify that the APL2 Application Environment module should be used when invoking APL2 from APL2PI.
- Specify the invocation options that should be used by APL2PI.

Performing Default Installation

If you do not want to customize, just do the default installation:

For APL2:

```
I5688228 INSTALL
```

For APL2 Application Environment:

```
I5688229 INSTALL
```

The default process assumes:

- The files are installed on your A-disk
- APLSCOPT TEXT is not included
- APL2's GDDM support is included (for Version 3 Release 1 of GDDM)
- APL2's SQL/DS support is included (for Version 3 Release 3 of SQL/DS)
- No 3800 fonts are generated

Note: Even though APL2's SQL/DS support is included in the default installation, the SQL/DS administrator must independently authorize the APL2 program and users to access the SQL/DS data, as described in “Preparing for Access of SQL/DS by APL2” on page 68.

Performing Customized Installation

To customize the installation:

1. Prepare for the customization process by invoking the APL2 installation EXEC with:

   For APL2:

   ```
   I5688228 LOAD
   ```

   For APL2 Application Environment:

   ```
   I5688229 LOAD
   ```

The LOAD parameter tells the EXEC to stop after the files are loaded from the tape.

Note: The APL2 installation EXEC defaults to the A-disk. If you want to load APL2 files from the distribution tape to a location other than the A-disk, see "Installing to a Different Disk and/or Retrieving Optional Material" on page 49.

2. If you want to customize the APL2 installation process during the initial installation, or if you are reinstalling APL2 to customize it, see Chapter 5, "Customizing APL2" on page 46.

3. If you want to customize system options during the initial installation, or if you are reinstalling APL2 to customize them, now is the time to edit and reassemble the AP2VIOPT or AP2VIOPE module. The APL2 system options are controlled by the module AP2VIOPT and APL2 Application Environment system options are controlled by the module AP2VIOPE. The procedure for modifying AP2VIOPT or AP2VIOPE is outlined under "Changing the APL2 System and Invocation Options" on page 62.

4. If you want to customize the APL2PI command options during the initial installation, or if you are reinstalling APL2 to customize them, now is the time to edit and reassemble the AP2XAPIC module. The procedure for modifying AP2XAPIC is outlined under "Changing the APL2PI Command Options" on page 63.

5. Gain access to GDDM, QMF, SQL/DS, and VS APL text decks.

   You can modify the installation process to perform link-edits that include new copies of GDDM, QMF, SQL/DS, and VS APL text decks (perhaps because you are using a different release of GDDM, QMF, or SQL/DS than the default, because maintenance has been applied to the decks or because you have modified the file types in your VS APL options module). At this point in the customized installation process, you should gain access to the required text decks for the products. As supplied by IBM, APL2 has prelinked into the APL2 load modules the stub text decks for GDDM Version 3 Release 1, QMF Version 3 Release 1, and SQL/DS Version 3 Release 3, and APL2 assumes the values in the default VS APL options module.

For GDDM	ADMRLIB TXTLIB
For QMF	DSQCIA TEXT DSQCLOD2 TEXT DSQCMCVP TEXT
For SQL/DS	ARIRVSTC TEXT
For VS APL	APLSCOPT TEXT

   If the listed text files are available when the installation EXEC I5688228 or I5688229 is used to generate APL2, the associated prelinked TEXT decks are automatically replaced by your local copies of the TEXT decks. However, to make I5688228 or I5688229 include the APLSCOPT TEXT deck, you'll have to customize the installation process as described in Chapter 4 in the section titled "Including a Modified APLSCOPT TEXT Deck" on page 53.

6. If you have SQL/DS, preprocess the SQL/DS interface module.

   Before AP 127 can access SQL/DS, the SQL/DS interface module must be preprocessed by the SQL/DS EXEC SQLPREP. In addition, authority must be granted to the users in order to run the APL2 SQL/DS packages, and use the database.

For complete details on these procedures, see Chapter 5, “Customizing APL2” on page 46.

7. Complete the installation by invoking the APL2 installation EXEC with:

   For APL2:

   ```
   I5688228 GENERATE
   ```

   For APL2 Application Environment:

   ```
   I5688229 GENERATE
   ```

   This command assumes the tape files are loaded and rebuilds the APL2 system from existing files. The functions performed by the installation EXECs are shown below.

For APL2 When building a system with I5688228 GENERATE, the I5688228 EXEC calls the AP2MAINT EXEC a number of times. In the default installation, the I5688228 EXEC:

- Invokes AP2MAINT CLEAN to erase any APL2 MODULE or APL2 LOADLIB from a previous installation.
- Invokes AP2MAINT APL2-PID to create a new APL2 MODULE and APL2 LOADLIB.
- Invokes AP2MAINT APL2SEG to attempt to save the APL2 saved segment.
- Invokes AP2MAINT CSVP-PID to create the global SVP LOADLIB and to save the AP2SM22 saved segment.
- Copies the AP2EXIT EXEC sample, if you do not already have one.
- Copies the DEFAULT VSAPLPR sample, if you do not already have one.
- Copies the LIBTAB APL2 sample, if you do not already have one.
- Copies the AP2MPSRV EXEC sample, if you do not already have one.
- Invokes the V5688228 EXEC to verify installation.

  The V5688228 EXEC loads APL2 and execute an APL program that identifies which components of the system are installed and whether these components appear to be functioning. The verification does not require any input from the terminal and displays messages to inform you of its progress. A sample output appears in Appendix C, “Sample APL2 Installation” on page 120.

For APL2 Application Environment When building a system with I5688229 GENERATE, the I5688229 EXEC calls the AP2MAINT EXEC a number of times. In the default installation, the I5688229 EXEC:

- Invokes AP2MAINT CLEANAE to erase any APL2AE MODULE or APL2AE LOADLIB.
- Invokes AP2MAINT APL2AE-PID to create a new APL2AE MODULE and APL2AE LOADLIB.
- Invokes AP2MAINT APL2AESEG to attempt to save the APL2AE saved segment.
- Invokes AP2MAINT CSVP-PID to create the global SVP LOADLIB and to save the AP2SM22 saved segment.
- Copies the AP2EXIT EXEC sample, if you do not already have one.
- Copies the LIBTAB APL2 sample, if you do not already have one.
- Copies the AP2MPSRV EXEC sample, if you do not already have one.
- Invokes the V5688229 EXEC to verify installation.

The V5688229 EXEC loads APL2 and executes an APL program that identifies which components of the system are installed and whether these components appear to be functioning. The verification does not require any input from the terminal and displays messages to inform you of its progress. A sample output appears in Appendix C, “Sample Application Environment Installation” on page 125.

The I5688228 or I5688229 EXEC exits with one of the return codes shown in Figure 2.

Figure 2. Return Codes Issued by the I5688228 or I5688229 EXEC

Code	Description
0	Product installed and required parts verified.
12	Installation failed for one of the following reasons: • Virtual machine size too small. • Tape error: unit not available, wrong tape mounted, or positioning error. • Required elements not available—AP2VUTIL module and AP2MAINT exec. • AP2MAINT exec failed, AP2MAINT return code given. • VMFPLC2 load or scan failed. • STATE failed for APL2 LOADLIB. • LINK to APL product disk failed. • Format of APL product disk failed. • Attempt to load contents of files 1 and 2 from installation tape to APL product disk failed. • 19E disk not linked in read/write mode.
24	Invalid or missing operand.
100	Only help text was produced.
111	You responded “no” or responded invalidly three times to the “Do you want to proceed...” prompt.
114	Incorrect password supplied for link to APL product disk.
222	The product files were loaded successfully, but an error occurred in the generation of the product.
999	Product installed but verification failed.

Note: If the installation or verification fails, you can still need to load APL2 maintenance tapes to complete your installation. Install any preventive service tapes that came with your APL2 package, and any PTFs you have ordered as a result of reading the PSP bucket. Follow the procedure given in “Maintaining APL2” on page 78.

Step 5–Review APL2 Installation (Required)

Do step 5 to verify that APL2 has been installed correctly. After V5688228 or V5688229 (the verification EXEC) completes, close the spooled console by issuing the following command:

```
CP SPOOL CONSOLE STOP CLOSE
```

For reference in investigation of any installation problems, make sure that you retain the printer file generated.

If it appears that APL2 has not been installed correctly, examine your console messages to verify successful completion of each installation step. If any installation step did not complete successfully, repeat the installation procedure from the beginning. If it appears that all installation steps were performed correctly and you are still having a problem, contact your IBM Branch Office or Systems Engineer.

Step 6–Copy Files to Production Disks (Optional)

Step 6 is not required, but we recommend that you perform it before APL2 is made available for production use. By copying the APL2 files to a different disk for production use, you can apply maintenance, recustomize your installation copy of APL2, and test to make sure that the new release works correctly without affecting active users of the production release. Only when the new release of APL2 has been successfully tested would you replace the production release of APL2 with the new one.

In the following examples of LINK and ACCESS commands, substitute these words with the appropriate word or number.

user ID The user ID that owns the production disk

vaddr1 The virtual device address in the directory

vaddr2 The virtual device address to be assigned to the minidisk

mode1 Filemode for disk containing the files to be copied. *mode1* can be * during installation. *mode1* must be the appropriate filemode if AP2MCOPY is used after applying maintenance (see step 9 on page 80).

mode2 Filemode for the target disk to which the files are copied.

To copy files to production disks:

1. Spool the console using the following command:

   ```
   CP SPOOL CONSOLE START
   ```

2. Review AP2EXIT, review LIBTAB, and copy APL2 execution files.

   Prior to making APL2 available for general use, you should review the AP2EXIT EXEC in Chapter 5, “Customizing APL2” on page 46.

   Review LIBTAB APL2. You'll need to modify the file LIBTAB APL2 if:

   - The public libraries are not accessed as an extension of the user's A-disk
   - Additional public libraries are to be defined
   - Additional private libraries are to be defined

   During the planning phase, if you designated a disk to contain the APL2 system files that are available to APL2 users at execution time, copy the system files

from your A-disk to that disk. Use the following commands and insert the user ID of the user who owns the disk to receive the files.

For APL2:

EXEC AP2MCOPY APL2 STATE

CP LINK *userid vaddr1 vaddr2* W

ACCESS *vaddr2 mode2*

EXEC AP2MCOPY APL2 CMDCALL COPYFILE *mode1* = = *mode2*

For APL2 Application Environment:

EXEC AP2MCOPY APL2AE STATE

CP LINK *userid vaddr1 vaddr2* W

ACCESS *vaddr2 mode2*

EXEC AP2MCOPY APL2AE CMDCALL COPYFILE *mode1* = = *mode2*

3. If you intend to make the APA printer fonts available, copy them to the disk containing other APA fonts for the system. No specific procedure is provided for this, but the files involved are all of those with a file type of `FONT38PP`, `FONT3812`, or `FONT3820`. The file names are all of the form `C18155`*xx*, `X18155`*xx*, or `T1200293`.

4. Copy global SVP files.

   If the global SVP has been defined and limited access to it is desired, you or the APL2 administrator should implement the AP2SVPEX command before making APL2 available for general use. See Chapter 5, “Customizing APL2” on page 46 for details regarding this installation exit.

   Copy the system files for the global SVP from the your A-disk to the A-disk for the global SVP user ID. Use the following commands and insert AP2SVP for the user ID unless a different one was selected.

   EXEC AP2MCOPY CSVP STATE

   CP LINK *userid vaddr1 vaddr2* W

   ACCESS *vaddr2 mode2*

   EXEC AP2MCOPY CSVP CMDCALL COPYFILE *mode1* = = *mode2*

5. Copy public libraries.

   Note: This does not apply to APL2 Application Environment.

   During the planning phase, if you designated a disk to contain the APL2 public libraries that are available to APL2 users at execution time, you should copy the distributed workspaces from your A-disk to that disk. Use the following commands and insert the user ID of the user who owns the disk that is to receive the public workspaces.

   EXEC AP2MCOPY WS STATE

   CP LINK *userid vaddr1 vaddr2* W

   ACCESS *vaddr2 mode2*

   EXEC AP2MCOPY WS CMDCALL COPYFILE *mode1* = = *mode2*

6. Back up the installation disk.

   IBM recommends that the disks used or affected by this installation be backed up at this time.

7. Run the verification procedure again.

   Note: The Installation Verification Procedure (CMSIVP) does not run against the production system unless the AP2VUTIL MODULE is accessed.

   To verify that all files have been correctly placed on the production disks, you should execute the verification procedure again from a user ID other than the user ID you used for installation. Since the verification EXEC is not copied to the production disks, you have to manually start APL2.

   In order to do this you should:

   - Log on to a different user ID, then link to and access the production disks
   - Use the RUN(CMSIVP) option when invoking APL2.

     Invoke APL2 by entering:

     ```
     APL2 SM(OFF) RUN(CMSIVP) INPUT (')OFF')
     ```

     Invoke APL2 Application Environment by entering:

     ```
     FILEDEF APLPRINT TERMINAL
     APL2AE RUN(CMSIVP)INPUT (')OFF')
     ```

     If you have not supplied an AP2SVPEX module on the global SVP service machine, then you must invoke APL2 with the ID parameter specifying a number other than zero if you want to verify the presence of the global SVP. Otherwise, the CMSIVP always shows the shared memory as local.

     You should see the messages normally displayed when APL2 is invoked:

     For APL2:

     ```
     APL2 2.2.00 (English)
     Version 2 Release 2

     CLEAR WS
     ```

     For APL2 Application Environment:

     ```
     APL2 2.2.00 (English)
     (Application Environment)
     Version 2 Release 2

     CLEAR WS
     ```

     Additionally, locally defined messages can follow APL2 version message if you have a GREETING APL2LANG file defined. The CMSIVP begins with the following message:

     ```
     APL2 INSTALLATION VERIFICATION PROGRAM
     5688-228, 5688-229

     THIS PROGRAM WILL DISPLAY INFORMATION DETERMINED
     ABOUT THE INSTALLATION OF THIS APL2 SYSTEM.
     .
     .
     ```

The text of the CMSIVP then appears. See Appendix C, “Sample Installation” on page 120 for an example. The following message signals the end of the verification function:

```
    END OF INSTALLATION VERIFICATION PROGRAM.
```

Finally, APL2 produces accounting messages and returns you to CMS:

```
CONNECTED    0.01.23
CPU TIME     0.00.16
R; T=16.03/24.82 11:18:57
```

Chapter 4. APL2 Invocation and Termination

APL2 is invoked from the host system environment by a command. The default command name is APL2, but your installation may have changed the name of the command. Whatever its name, you may either:

- Manually enter it after the host system is initialized.
- Execute an EXEC that invokes the APL2 command.
- Have the command automatically invoked when you enter CMS. This is normally done through a PROFILE EXEC.

The)*OFF* or)*CONTINUE* system command terminates your APL2 session and allows either you or the invoking EXEC to continue with your CMS session.

Appendix D, “APL2 System Options” on page 129 contains a summary of invocation options.

Appendix E, “APL2 Files and Data Sets” on page 133 contains AP2EXIT EXEC samples.

For a description of the underlying system during invocation and termination, see *VM/SP CMS User's Guide*.

Starting APL2—The APL2 Command

You can invoke APL2 with the following command:

For APL2

APL2 *option1, option2, ..., option*

For APL2 Application Environment

APL2AE *option1, option2, ..., option*

where:

option Is one of the keyword parameters used to tailor the APL2 environment for a particular APL2 session. See “Continuing with the Invocation Options” on page 26 for the invocation options list.

Before invoking APL2, you should have:

- A 1-megabyte virtual machine if APL is installed in shared storage.

  You need more if your site has not installed APL2 in shared storage or if you are using large applications.

- An AP2EXIT EXEC

  This exit is invoked by APL2 when you invoke and leave APL2. The parameters passed to it are defined in the sample provided.

© Copyright IBM Corp. 1984, 1994

- A LIBTAB APL2 file

  A sample is provided with the licensed program and may have been modified.

A complete list of DDNAMEs recognized by APL2 under CMS can be found in Appendix E, “APL2 Files and Data Sets” on page 133.

Continuing with the Invocation Options

Invocation options entered via keyword parameters to the APL2 command are available to tailor the APL2 environment for a particular APL2 session. The invocation options are summarized alphabetically by keyword in Figure 3 on page 27 and described in detail starting on page 28. To find out quickly what the option keywords are, enter the following: `HELP APL2`

Each keyword option is usually followed by a value in parentheses. Default values are supplied for the options, and these are generally satisfactory for beginning APL2 users. You may, however, want to override some of the options to customize the APL2 session to meet your needs. To override one or more options, enter the APL2 command followed by the keyword and value of the option(s) you want to override. For example, you enter the following command to invoke APL2 when the APL2 session manager is to be suppressed: `APL2 SMAPL(OFF)`

Keyword options can be specified in any order after the APL2 command, and the keywords can be abbreviated. The abbreviation must be at least the first two characters of the keyword.

If the same keyword appears more than once in the APL2 command, the value of the last one entered is used. For example, the following APL2 command initiates an APL2 session without the session manager: `APL2 SMAPL(ON) ID(1007) SMAPL(OFF)`

The DEBUG, SYSDEBUG, and TRACE options can be specified as positive or negative integers. The system treats them as binary flags and turns the corresponding flags on if the integer is positive or turns them off if the integer is negative. For example:

`DEBUG(7)` A value of 7 is treated as binary 1 1 1 and specifies setting `DEBUG (4 2 1)` on.

`DEBUG(-5)` A value of `-5` is treated as binary -(1 0 1) and specifies setting `DEBUG(4 1)` off.

Successive specifications have the effect of successively turning the corresponding flags on and off. For example:

`DEBUG(6) DEBUG(-3)` First, the 4 and 2 DEBUG flags are set; next, the 2 and 1 DEBUG flags are turned off. The result is a setting of `DEBUG(4)`.

The defaults supplied with the APL2 Licensed Program are given with the detailed descriptions of the options beginning on page 28. Your installation may have changed some or all of these defaults. Check with your system administrator to find out which of these options may have been changed during the installation of APL2 at your site.

Figure 3. Summary of APL2 Invocation Options

Keyword	Description
APNAMES(*name*[('*string*')]...)	Specifies the auxiliary processors to be used in addition to those automatically loaded.
CASE(*n*)	Identifies the cases of the alphabets to be used when APL2 accepts, returns, or displays the names of the APL objects.
CODE(-1)	Same as TERMCODE option.
DATEFORM(ISO|ES|EU)	Specifies the format for time and date stamps.
DBCS(TRY|ON|OFF|*nnn*)	Indicates whether mixed APL2/DBCS is supported for I/O during the APL2 session.
DEBUG(*nnn*)	Alters normal error recovery actions of APL2 for debugging purposes.
DSOPEN(*device-token*)	Specifies a device-token to be passed to GDDM (if it is available) on a DSOPEN call.
EXCLUDE(*name...*)	Specifies the auxiliary processors you do not want loaded into your session.
FREESIZE(*size*)	Specifies a minimum limit on the amount of virtual storage not used for the active workspace or shared variables.
HILIGHT(*setting*)	Specifies whether input, output, or both are to be highlighted on the screen.
ID(*nnnnnnn*)	Specifies an identifier number to be associated with the current APL2 session.
INPUT('*string*' ...)	Specifies one or more input lines to be given to APL2 upon invocation.
NLT(*language*)	Allows users to override the installation default for the national language in their invocation EXEC or CLIST, and allows the language to be set before APL2 is running.
PROFILE(*name*)	Identifies the name of the APL2 session manager profile to be loaded on invocation.
QUIET[(ON|OFF)]	Prevents APL2 from displaying output until it prompts for input.
RUN(['*locator*'] *function*)	Specifies name of a niladic function in a namespace as a part of entry to APL2.
SHRSIZE(*size*)	Specifies the amount of virtual storage to be reserved for the shared variable processor.
SMAPL(TRY|ON|OFF|*nnnn*)	Specifies whether or not the APL2 session manager is to be used during your APL2 session or whether the interpreter should share a variable with processor *nnnn*.
SVMAX(*nnnnn*)	Specifies the maximum number of shared variables that you can concurrently share.
SYSDEBUG(*nnn*)	Specifies special debugging settings for use by system programmers.
TERMCODE(-1)	Specifies controlled invocation.
TRACE(*nnn*)	Provides special debugging aids for use by system programmers.
WSSIZE(*size*)	Specifies the amount of virtual storage in your CMS virtual machine to be reserved for your active workspace.
XA(*nn*)	Identifies whether working storage in an XA or ESA system should be allocated above or below the 16-megabyte address boundary.

Description of Invocation Options

This section lists the invocation options in alphabetic order by keyword. The headings show the syntax of the options; the descriptions explain how to use each option and what values to specify.

Three options allow you to specify an integer *size* value. These options are:

```
FREESIZE(size)
SHRSIZE(size)
WSSIZE(size)
```

Unless otherwise restricted by the maximum value allowed for the option, *size* can be specified as bytes, kilobytes, or megabytes. To specify kilobytes or megabytes, follow *size* with a K or M, respectively. To specify a size in bytes, enter a number only. For example, all the following are valid values for the FREESIZE option:

```
FREESIZE(2048000)
FREESIZE(2000K)
FREESIZE(2M)
```

You may also express *size* as a percentage. For example, to reserve 50% of the virtual machine size for your workspace, you could specify:

```
WSSIZE(50%)
```

Your installation of APL2 may provide defaults for each of the invocation options. Your installation may also prevent you from changing some of these options by providing system overrides for them. Check with your system administrator to find out which, if any, options have system overrides.

APNAMES(*name*[*('string')*]...)

The APNAMES option identifies auxiliary processors not automatically available but which you plan to use during your APL2 session. Processors named with this option can be those distributed with the APL2 Licensed Program or those written by your system programming staff.

name

Name of the load module for the processor you want available. More than one name can be specified. For example:

```
APNAMES(USERAP1 USERAP2)
```

Your installation normally provides resident auxiliary processors that are automatically available to your session. If you specify a name in this option that is the same as the name of a resident processor, the resident version is ignored and a load module with the specified name is searched for and loaded. If you specify an auxiliary processor that uses the same number as the resident auxiliary processor, the specified auxiliary processor should have the same name as the resident, or you must use the exclude option to exclude the resident. For example:

```
APN(AO121) EXC(AP2X121)
```

Figure 4 on page 30 contains the names of the auxiliary processor load modules supplied with the APL2 Licensed Program.

Under CMS, the following are searched to locate the specified module name:

- Nucleus extensions

  One way to load a module as a CMS nucleus extension is the CMS NUCXLOAD command.

- Members of a CMS LOADLIB defined by the CMS FILEDEF command as AP2LOAD. For information on creating auxiliary processor LOADLIB files, see *APL2/370 Installation and Customization under CMS*.

`('`*string*`')`
Optional character string of parameters to be passed to the auxiliary processor. For example:

```
APNAMES(USERAP1 USERAP2('EBCD') USERAP3('L'))
```

Most auxiliary processors distributed with APL2 ignore any character strings specified as parameters. The *string* parameter may be used with user-written auxiliary processors and with AP 119 and AP 127.

AP 119 Access: The following options can be specified with the APNAMES parameter:

SERVPORT(*nnn***)**
Normally, the local port server is listening on port number 31415. If the port server is using a different port number, this parameter must be used to allow AP 119 to communicate with the server. Allowable values are 256 to 65535.

LISTEN(*nnn***)**
AP 119 does not normally open a listening connection until a variable is offered to a remote processor. If you would like AP 119 to open a listening connection on startup, use this parameter. AP 119 attempts to use the number specified as its listening port. If this number is unavailable, the listening connection is not started. Allowable values are 256 to 65535, or 0 to let TCP/IP assign an arbitrary number.

TCPID('*ccc***')**
AP 119 expects the name of the local TCP/IP machine. If the name is different, this parameter must be used to allow AP 119 to successfully communicate with TCP/IP.

This is an example of using the startup parameters:

```
APL2 APNAMES(AP2X119(SERVPORT(1234) LISTEN(2345) TCPID('TCPTEST')))
```

If no options are specified using APNAMES, the default is:

```
SERVPORT(31415) TCPID('TCPIP')
```

For further information, see “Running the Port Server” on page 75.

AP 127 Access: You can use APNAMES to override some default values for AP 127:

```
APNAMES(AP2X127(ISOL(level)))
```

level
Character string that specifies the starting isolation level. Must be RR or CS. The default is **RR**.

The auxiliary processor AP 127 needs to be specified on APL2 invocation only if you want to override one or more of these defaults. For any keyword not specified, the defaults from the installation options module is used.

Figure 4. Module Names of Auxiliary Processors Supplied with APL2

Auxiliary Processor	Module Name
AP 100	AP2V100
AP 101	AP2V101
AP 102	AP2V102
AP 110	AP2V110
AP 111	AP2V111
AP 119	AP2X119
AP 120	AP2X120
AP 121	AP2X121
AP 123	AP2V123
AP 124	AP2X124
AP 126	AP2X126
AP 127	AP2X127
AP 211	AP2X211

CASE(*n*)

Identifies the cases of the alphabets to be used when APL2 returns or displays the names of APL objects.

n One of the following:

0 Both lowercase and underbarred characters are valid and synonymous when evaluating names. Primitives that return names as results (⎕*NL*, ⎕*CR*, ⎕*FX*, ⎕*SVQ*, ⎕*TF*), and system commands and messages that produce names, produce them using the underbarred rather than the lowercase alphabet.

1 Both lowercase and underbarred characters are valid and synonymous when evaluating names. Primitives that return names as results, and system commands and messages that produce names, produce them using the lowercase rather than the underbarred alphabet.

2 In general, underbarred characters are not valid in names and are not accepted or produced in system functions, commands, or messages. Underbarred letters in the arguments to)*COPY*,)*PCOPY*,)*MCOPY*, and)*IN* are accepted as lowercase letters to aid migration.

Note: The response to)*ERASE*, given underbarred letters, is the underbarred letters.

The default shipped by IBM is CASE(1). Your installation may change the default or specify an overriding value.

The CASE(*n*) invocation parameter does **not** apply to all work done by you during the APL2 session. Instead, it is interpreted as an implicit parameter of a)*CLEAR* command. CASE is, in fact, a workspace attribute in all workspaces. (Any workspaces created by APL2 release 1 have an attribute of 0; and the default for releases 2 and 3 was also 0.) The workspace attribute is saved with the workspace and is not changed when the workspace is loaded.

Objects may be copied into the active workspace using)*COPY*,)*PCOPY*,)*MCOPY* or)*IN* without affecting the workspace attribute. The names of copied objects, as well as names referred to by copied functions or defined operators, are converted appropriately.

Note: Literal strings and comments within functions, and the content of indirect copy lists, are **not** converted.

No matter what CASE setting or workspace attribute is in effect, shared variable names containing underbarred characters are never transferred from the APL2 interpreter to the SVP. The underbarred characters are instead translated to lowercase. Names returned to the interpreter from the SVP are handled and displayed in accordance with the workspace attribute.

CODE(-1)

Used to request controlled invocation. CODE is a synonym for TERMCODE.

CODE can only be used to request controlled invocation. CODE(-1) is the only value accepted. Under APL2 Application Environment, TERMCODE(-1) is assumed.

APL2 avoids using the terminal if controlled invocation is used. Instead, APL2 redirects input and output requests to files, much as if it were running on a disconnected virtual machine. Controlled invocation is intended for use by applications that are themselves using the terminal and do not want APL2 to interfere with their terminal input and output.

DATEFORM(ISO|US|EU)

Specifies the format for date and time stamps displayed during operations, such as)*LOAD* and)*SAVE*.

ISO
International Standards Organization convention, which follows the format *yyyy-mm-dd hh.mm.ss*.

ISO is the IBM-supplied default.

US
United States preferred convention, which follows the format *mm/dd/yyyy hh.mm.ss*.

EU
European convention, which follows the format *dd.mm.yyyy hh.mm.ss*.

For example, to display the European format, you specify: `DATEFORM(EU)`

The date and time are formatted as: `27.03.1992 11.30.52 (`*GMT*`+1)`

Note: The APL2 date and time display is always followed by the user's time zone offset from Greenwich Mean Time (GMT).

DBCS(TRY|ON|OFF|*nnn*)

The DBCS invocation option is used to tell APL2 what double-byte character set (DBCS) data to display and input during the APL2 session. It also sets the default DBCS number for the DBCS translation option of various auxiliary processors and external functions.

Many languages have more characters than can fit in the APL2 atomic vector, □*AV*. Japanese, Korean, and Chinese are examples. IBM has defined DBCS to represent these languages. These characters can be displayed on displays that support the DBCS, such as the IBM 5550 multistation. Each DBCS character occupies two columns on the screen. DBCS characters and characters from □*AV* may be mixed on the screen.

Note: APL2 supports the display of DBCS only through the session manager, AP 126, and)EDITOR 2. GDDM is required. For testing purposes, GDDM can provide limited, emulated DBCS support on displays that do not actually support DBCS.

APL2 also supports DBCS in APLIN and APLPRINT files.

When DBCS data is being processed, the DBCS invocation option can control the identification of the DBCS used. In particular, a user with a display that does not support real DBCS or with a display supporting one DBCS can simulate operation on a display with a different DBCS.

If the DBCS invocation option is not used, APL2 determines whether or not mixed APL2/DBCS data is to be displayed based on the presence of DBCS support as reported by GDDM. If GDDM reports that the display has a DBCS, or that GDDM

emulated DBCS support is available, then APL2 allows mixed DBCS data. For displays that have real DBCS support, the DBCS ID defaults to the CPGID of the DBCS display. For GDDM emulated DBCS support, the DBCS ID defaults to 0.

DBCS(TRY) If GDDM indicates that the display has a DBCS, then activate DBCS support and use the CPGID as the DBCS identifier.

If GDDM indicates that emulated DBCS support is being used, then treat as DBCS(0).

If GDDM is not available, or it indicates that no DBCS support is available, then treat as DBCS(OFF).

Note: DBCS(TRY) is the default in the installation options module as distributed by IBM, but your installation may choose a different default.

DBCS(ON) Behaves like DBCS(TRY), except this ends the APL2 session if DBCS is not available rather than treating as DBCS(OFF).

DBCS(OFF)

- Substitute '⍵' for any DBCS characters on terminal output.
- Do not accept DBCS on terminal input.
- Use 0 as the DBCS identifier when auxiliary processors and function routines convert DBCS data to APL data.
- Permit any DBCS identifier when auxiliary processors and function routines convert APL data to DBCS data.

DBCS(*nnn*) Use the decimal value *nnnnn* as the DBCS identifier and activate DBCS display support. The value of *nnn* must be between 0 and 32,767 inclusive.

Only APL data with the specified DBCS identifier can be converted to DBCS data by auxiliary processors, function routines, or terminal output processors.

Note: DBCS(0) is not equivalent to DBCS(OFF), even for auxiliary processors or function routines.

This invocation option may be changed during the APL2 session by either the *OPTION* external function or *)CHECK SYSTEM DBCS*. If this is done, APL2 display support is restarted. For the session manager, this means that any in-storage log file is discarded.

DEBUG(*nnn*)

Alters normal error recovery actions taken by APL2. It can assist in debugging errors.

Note: Your installation may supply one or more debug settings that you cannot override.

nnn

One or more numbers (positive or negative) of the debug settings you want to change. A positive number turns on one or more settings, whereas a negative number turns off one or more settings. Several settings can be turned on or turned off with a single number equal to the sum of the numbers for the individual settings. For example, to specify debug settings 2 and 4, specify either:

```
DEBUG (2 4)
```
or
```
DEBUG (6)
```

To turn off setting 4, specify:

```
DEBUG (-4)
```

IBM-supplied default value: 0

Your installation may provide a default DEBUG value other than 0.

The debug settings and their meanings are:

1—MSG Displays secondary error messages without prompting.

Note: Use of this setting produces messages for exceptional conditions that are not necessarily error conditions. For example, end of file may or may not be an error, depending on the situation. An error message is displayed immediately, before an attempt to determine if the condition is acceptable.

2—ECHO Causes all input to APL2 from the AP 101 input stack to be "echoed" (displayed) at your terminal. Normally, input from the AP 101 stack is not displayed at the terminal as it is executed.

4—XDUMP Default setting; provides more complete dumps. Typical dumps are about 50 pages long. Under this setting, they may be 200 to 500 pages or more, depending on the workspace size. For a further explanation of APL2 storage dumps, see *APL2/370 Diagnosis Guide*.

32—MSGID Provides the message identifier number, along with the text of any message produced by APL2. For example, the following message:

```
DOMAIN ERROR
```

displays with DEBUG(32) as:

```
AP2IAPL254 DOMAIN ERROR
```

64—NOLX Suppresses execution of ⎕*LX*, the latent expression, when workspaces are loaded with the)*LOAD* command.

128—NOQUEMSG

Discards secondary messages rather than queuing them. While 1—MSG is turned on, secondary messages are immediately displayed rather than queued, so the setting of this flag is irrelevant.

DSOPEN(*device-token***)**

Specifies a value to be passed to the Graphical Data Display Manager (GDDM) as the device-token parameter on a GDDM DSOPEN call. This option permits more precise declaration of the terminal type when GDDM cannot automatically determine the correct terminal characteristics.

You must specify a device token with DSOPEN if you want to access GDDM through AP 126 while running APL2 in a CMS disconnected virtual machine. Otherwise AP 126 issues GDDM calls during AP 126 initialization that tell GDDM to use device characteristics of the current terminal type. Since no terminal exists in a CMS disconnected virtual machine, GDDM may hang or otherwise fail to initialize.

You also may need to specify a device token with DSOPEN if you are using a 24-line display screen under TSO. In this case, GDDM cannot determine whether the terminal type is a 3277 or a 3278. In such cases, GDDM simply assumes some terminal type that may not be correct. For more information about terminal types, see the description of the TERMCODE option below.

This option is meaningful only when the APL2 session manager, AP 126, or)*EDITOR 2* is used.

device-token

Selected device tokens supplied by GDDM include:

ADMK772A 3277 model 2 with APL

ADMK782A 3278 model 2 with APL

For more values, see the description of the device tokens in *Graphical Data Display Manager (GDDM) Base Programming Reference*.

An example of using the DSOPEN option is shown below:

```
DSOPEN(ADMK782A)
```

EXCLUDE(*name...***)**

Specifies the module names of auxiliary processors normally available but which you do not want loaded at the start of your APL2 session. For example, to exclude AP 119, the data file processor, from being loaded when you invoke APL2, include the EXCLUDE option in the APL2 command as follows:

```
EXCLUDE(AP2X119)
```

Figure 4 on page 30 contains the load module names of auxiliary processors supplied with the APL2 Licensed Program. Check with your system administrator to find out the names of any additional processors that may be automatically available at your installation.

Caution: Excluding auxiliary processors can reduce storage requirements for APL2. However, excluding certain IBM auxiliary processors may cause problems

because of internal APL2 dependencies on these processors. The following auxiliary processors may be safely excluded, if users do not need them: AP 119, AP 120, AP 121, AP 123, AP 124, AP 127, AP 210, and AP 211.

FREESIZE(*size*)

Specifies the amount of virtual storage that must be unused after space is allocated for the active workspace, shared variables, and other areas allocated during invocation. These areas include modules and work areas used by other products such as VSAM and GDDM. FREESIZE may be needed during the session for the APL2 interpreter module, auxiliary processor modules, access method buffers, dynamically loaded modules, storage control blocks, and program products such as GDDM.

size

An integer expressed in bytes, kilobytes, or megabytes. For example:

```
FREESIZE(65536)
     or
FREESIZE(64K)
```

A percentage of your virtual machine size. For example: `FREESIZE(10%)`

Your installation provides a default value for FREESIZE. If the minimum amount of FREESIZE is not available, APL2 invocation fails.

Because of the number of variable factors associated with APL2 storage allocations, you may find it better to omit the FREESIZE option and use the WSSIZE and SHRSIZE options to reduce the amount of space allocated to the active workspace and shared variables. If you cannot reasonably reduce WSSIZE and SHRSIZE and you are having virtual storage problems, you must increase your CMS virtual machine size.

HILIGHT(*setting*)

Specifies whether input, output, or both are to be highlighted on the screen.

Note: This option applies only when the APL2 session manager is used.

setting

One of the following:

INPUT Highlight only input lines. (default)

OUTPUT Highlight only output lines.

ON Highlight all lines.

OFF Highlight no lines.

For example, to have no lines highlighted during your APL2 session, specify:

```
HILIGHT(OFF)
```

If the session manager is not used, the setting of the CP TERMINAL HILIGHT command determines whether or not user input lines are highlighted.

ID(*nnnnnnn***)**

Specifies a numeric identifier to be associated with your current APL2 session. The number becomes the first item in the system variable ⎕*AI* (see *APL2 Programming: Language Reference*). This number identifies your:

- Default library for saved workspaces (including any *CONTINUE* workspace)
- Default library for APL data files
- Library for APL2 session manager log files

The number you enter in the parameter to the ID option is also used to identify you as a possible share partner for shared variable communication between users on the same system. The value of the ID parameter does not affect a user's ability to share variables across systems. TCP/IP profile files are used to define potential cross system partners. (See *APL2/370 Programming: System Services Reference*.)

nnnnnnn
: Should be at least 1000 to avoid conflict with auxiliary processor identifiers (which are usually 100 through 999). The number cannot be greater than 9999999.

For example, to identify yourself as user 1234 when you invoke APL2, specify:

```
ID(1234)
```

If two or more APL2 users or auxiliary processors have the same ID number, only one of these users can communicate with the shared variable processor (SVP).

Your installation has the ability to set or change your ID and to override any value you provide. It also has the ability to restrict the use of user-to-user shared variables on the same system. Check with your system administrator for APL2 ID requirements at your site.

Unless your installation overrides your specification, if you do not specify an ID (or if you specify ID(0)) when you invoke APL2, you are not permitted to share variables with other users except using cross systems shared variables.

INPUT('*string*'....)

Specifies one or more input lines used when APL2 is invoked. These lines are processed before APL2 requests input from any other input source—the program stack, the terminal, or, in the case of batch or controlled invocation, an input file (see TERMCODE below).

'*string*'...
: Each character string represents a line of input to APL2. Strings are separated by one or more blanks or commas.

  Each string is enclosed in a pair of quotation marks. If quotation marks are a part of an expression, a *pair* of single quotation marks must be entered for each *single* quotation mark in the expression.

  The example below illustrates the use of the INPUT option to load an APL2 workspace named *PAYROLL*, and then execute a function named *START* with an argument '*ABC*'.

```
INPUT(')LOAD PAYROLL'  ' START ''ABC''')
```

Note: If you do not specify the INPUT option when invoking APL2, the APL2 session begins with the *CLEAR WS* message or the loading of a *CONTINUE* workspace. If you have a damaged *CONTINUE* workspace and want to suppress the automatic loading of it, you can specify the INPUT invocation option with no data:

```
INPUT('   ').
```

NLT(*language*)

Allows users to override the installation default for the national language in their invocation EXEC or CLIST, and allows the language to be set before APL2 is running. For example:

```
APL2 NLT(ESPANOL)
```

For a list of the national languages supported by APL2, see *APL2/370 Programming: System Services Reference*.

PROFILE(*name***)** **(SMAPL Only)**

Specifies the name of the file containing the profile to be used by the APL2 session manager. During its initialization, the session manager executes a PROFILE LOAD command using the name that you specify. (See the description of the PROFILE LOAD session manager command in *APL2/370 Programming: System Services Reference*.)

name
: File name of the profile to be loaded. The file type of the named profile must be VSAPLPR.

If you are using the session manager without specifying the PROFILE option, the profile named DEFAULT is loaded. (If more than one DEFAULT profile exists, the host system search order determines which one is loaded.)

If you are not using the session manager, the PROFILE option is ignored.

If you specify PROFILE( )—no profile name—the session manager does not execute a PROFILE LOAD command.

For further information on session manager profiles, see *APL2/370 Programming: System Services Reference*.

QUIET[**(ON|OFF)**]

ON
Prevents APL2 from displaying any output until APL2 prompts for input.

OFF
Permits APL2 to display output when APL2 prompts for input. It is the default.

For instance, if you use the INPUT option and the QUIET option, you can load a workspace and start an application without displaying the APL2 initiation messages that would normally be displayed. If that application uses AP 101 to stack an *)OFF* system command, APL2 termination messages are suppressed.

Note: The QUIET option only applies to output from the APL2 language processor. All other output, such as that from AP 126, is not suppressed.

If you are using the session manager, the QUIET option does not suppress the initial display of the session manager screen if the session manager profile includes the DISPLAY(ON) command. To suppress initial display of the session manager screen, specify the PROFILE() option or load a session manager profile that does not include the DISPLAY(ON) command.

Use the external function OPTION to change the QUIET setting. See *APL2/370 Programming: Using the Supplied Routines* for more information.

RUN([*'locator'*] *function*)

This option is designed to simplify invocation of an external function as a part of entry to APL2.

locator
Optional information to be used by processor 11 to locate the external function. If provided, this must be a character string that can be used as the first item of the left argument to *⎕NA* for processor 11. When this information is provided, the external function must either reside in a namespace or be self-describing. The possible formats are:

'member'
The name of a nucleus extension, MODULE file, or TEXT file containing the external function.

'library.member'
library is the file name of a LOADLIB file, and *member* is the name of a member in the library containing the function.

If *locator* is omitted, the function must be described in a NAMES file entry for the *function* using the defaults NAMES files for the APL2 session.

function
A 1 to 8 character APL name containing only alphanumeric characters (no Δ, _, ¯, or underbarred characters.) Note that lower case letters are converted to upper case while processing the invocation RUN option.

For more information, see *APL2/370 Programming: System Services Reference*. The RUN option produces a pair of statements that are executed before anything provided by the INPUT option, the stack, of the APLIN file. The first statement is a `⎕NA` for the requested function, and the second is a simple niladic invocation of it.

RUN provides only minimum control of application invocation. For more complex situations, use the INPUT option or provide an APLIN file.

SHRSIZE(*size*)

Specifies the amount of virtual storage to be reserved for the shared variable processor. The size of shared variable storage limits the volume of data that can be assigned to shared variables at any given time.

size
An integer expressed as bytes, kilobytes, or megabytes, as in the following example:

```
SHRSIZE(256000)
      or
SHRSIZE(250K)
```

A percentage of virtual machine size. For example: `SHRSIZE(5%)`

Minimum size: 16K bytes
Maximum size: Less than 16 megabytes
IBM-supplied default: 40K bytes

These sizes may be changed by your installation.

Note: Because of rounding, the actual SHRSIZE size assigned to your APL2 session may be slightly different from what you specify. Issue the `)QUOTA` system command to display the actual share size used. Your installation may allocate additional space for shared storage to support user-to-user shared variables. This space is not reported in the `)QUOTA` command. The `)QUOTA` command is described in *APL2 Programming: Language Reference*.

Typically, SHRSIZE values should be smaller than WSSIZE.

The SHRSIZE and WSSIZE come out of your virtual machine.

SMAPL(TRY|ON|OFF|*nnnn*)

Indicates whether you want to use the APL2 session manager for your APL2 session. The session manager requires GDDM and an IBM 3270 series display station.

TRY
Invokes the APL2 session manager if it is available.

TRY is the default setting.

ON
Invokes the APL2 session manager.

If the session manager is not available and you specify SMAPL(ON), you receive a message, and your APL2 session is not initialized.

OFF
Does not invoke the APL2 session manager.

If you specify SMAPL(OFF), APL2 uses the standard input/output protocol of the host system—CMS. If you do not use the APL2 session manager, but GDDM is available, you can still use the Editor 2 and AP 126, the GDDM Processor.

nnnn
If the SMAPL value is numeric, it indicates that the Shared Variable Interpreter Interface is to be used. The APL2 interpreter shares a variable with processor *nnnn*. All subsequent input and output for the interpreter is performed through the shared variable.

Note: Your terminal appears to become inactive.

The Shared Variable Interpreter Interface is designed to allow the interpreter to be controlled by another user ID, using an APL2 session manager or an APL2 function. The other ID can be on the same VM system, or can be on a different system, even a completely different type of system, provided the two systems are connected through TCP/IP. Once a variable is shared with the interpreter, the interpreter sends messages, arrays, and requests for input through the shared variable.

For further details about the shared variable interpreter's protocols, consult *APL2/370 Programming: System Services Reference*. For information about how to interactively communicate with the interpreter from another CMS or TSO session, consult the discussion of the `RAPL2` function in *APL2/370 Programming: Using the Supplied Routines*.

SVMAX(*nnnnn*)

Specifies the maximum number of shared variables you can share concurrently.

nnnnn
Must be a positive integer.

Minimum value: 4
IBM-supplied default: 88
Maximum value: 32767

The default value may have been changed during installation.

To display the current value of this option after you invoke APL2, issue the)*QUOTA* system command.)*QUOTA* is described in *APL2 Programming: Language Reference*.

Comment: The maximum number of shared variables is also limited by the size of shared storage. Each shared variable requires at least 128 bytes of space in shared storage.

SYSDEBUG(*nnn*)

Establishes special debug settings for your APL2 session. These settings provide information beyond that provided by the DEBUG option. They are intended for use by system programmers in diagnosing system or internal APL2 problems or problems with writing an auxiliary processor. They are not for use during normal APL2 operation; their use can significantly degrade APL2 performance.

For a description of the SYSDEBUG option, see *APL2/370 Diagnosis Guide*.

TERMCODE(-1)

Used to request controlled invocation. TERMCODE is a synonym for CODE.

TERMCODE can only be used to request controlled invocation. TERMCODE(-1) is the only value accepted. Under APL2 Application Environment, TERMCODE(-1) is assumed.

Except when an APL2 application explicitly requests full-screen I/O via AP 124 or AP 126, APL2 avoids using the terminal if controlled invocation is used. Instead, APL2 redirects input and output requests to files, much as if it were running on a disconnected virtual machine. Controlled invocation is intended for use by applications that are themselves using the terminal and do not want APL2 to interfere with their terminal input and output.

TRACE(*nnn*)

Provides system diagnostic output during your APL2 session. The option is intended for use by system programmers in diagnosing system or internal APL2 problems. It is not for use during normal APL2 operation and can significantly degrade APL2 performance; its use also interferes with normal terminal communication.

For a description of the TRACE option, see *APL2/370 Diagnosis Guide*.

WSSIZE(*size*)

Specifies the amount of *contiguous* virtual storage in your virtual machine to be reserved for your active workspace.

size

Integer expressed in bytes, kilobytes, or megabytes, as in the following:

```
WSSIZE(1048576)

WSSIZE(1024K)

WSSIZE(1M)
```

The size can also be expressed as a percentage of the CMS virtual machine size. For example:

```
WSSIZE(40%)
```

The size may also be specified as a negative amount. In that case, the workspace allocated is the largest size possible greater than the specified amount:

```
WSSIZE(-1M)

WSSIZE(¯1024K)

WSSIZE(-40%)
```

Minimum size: 16K bytes
Maximum size: APL supports up to 999 megabytes under CMS. The available storage may be limited further by your system.
IBM-supplied default: 25% of virtual machine size (CMS)

You can issue the)*QUOTA* system command after invoking APL2 to display the default size of the active workspace. The)*QUOTA* command is described in *APL2 Programming: Language Reference*.

XA(*nn*)

Identifies the address range from which working storage should be allocated on XA or ESA systems.

nn One of the following:

24 Working storage should only be allocated below the 16-megabyte boundary.

31 Working storage may be allocated either above or below the 16-megabyte boundary.

If no XA() value is specified, then working storage is allocated where ever it is available. On XA or ESA systems, this may be above the 16-megabyte boundary.

IBM does not supply a default setting for XA(). Your installation may supply a default, or specify an overriding value.

If you are on a system that does not support XA or ESA mode addressing, specifying XA(31) causes invocation to fail.

Even when XA(31) is specified or defaulted, some pieces of working storage must be allocated below the 16-megabyte line due to macro or access method restrictions.

Session Termination

When you enter the system command)*OFF* or)*CONTINUE*, APL2 calls the AP2EXIT EXEC again and executes the commands that reset your terminal for the continuation of your CMS session. Control then returns to CMS or to the program or the EXEC that invoked APL2.

APL2 returns return code 0 if an)*OFF* or)*CONTINUE* system command successfully ended the session.

APL2 returns return code 4 if all sources of input are exhausted but no)*OFF* or)*CONTINUE* system command has been encountered. This situation may occur, for instance, under CMS batch, when there is no system command to end the APL2 session in the input stream. APL2 saves the active workspace in *CONTINUE* before terminating if this situation does occur.

APL2 returns return code 8 if an error occurs that prevents APL2 from initializing. This situation may occur, for instance, if there is lack of sufficient storage.

APL2 returns return code 16 if an error occurs after initialization that forces APL2 to terminate further processing abnormally. This situation may occur, for instance, when APL2 has part of its program storage over-written by another program.

Any input lines remaining on the AP 101 alternate input stack after)*OFF* or)*CONTINUE* are available to CMS, unless APL2 was invoked with TERMCODE(-1).

Appendix E, “APL2 Files and Data Sets,” contains a sample AP2EXIT EXEC.

Session Interruption

If a system interrupt occurs during your session and you become disconnected, you can attempt to save any new work since your last)*SAVE* by reconnecting to your virtual machine. Do the following:

1. Log on to the same type of terminal you were using before. If you log on to a different type of terminal, APL2 may be unable to properly process your terminal input and output.
2. If you are reconnected to control processor (CP), a message similar to the following appears: `RECONNECTED AT 07:47:31 PDT MONDAY 07/12/87`

   Your installation may impose a time limit on how long you can be disconnected. If you do not reconnect within the time limit, CP logs off your virtual machine, and you lose the contents of your active workspace. When logging back on, you see the usual initialization messages instead of the RECONNECTED message.
3. Restore the CP settings established by AP2EXIT EXEC before you attempt to reinvoke APL2. (When you reconnect, AP2EXIT EXEC is not executed.)

   The CP settings in the AP2EXIT EXEC supplied with APL2 for a display terminal are:

   ```
   TERMINAL APL ON
   TERMINAL ATTN OFF (for non-3270 terminals)
   ```
4. Enter the CP command:

   ```
   BEGIN (or B)
   ```

   This command takes you from CP mode and returns you to your APL2 session.

Chapter 5. Customizing APL2

You can customize APL2 to meet the needs you identified in your installation plan. This chapter discusses the various ways in which you can modify APL2 to meet your installation requirements.

Customizing IBM-Supplied Sample Files

IBM supplies samples for a number of the files used by APL2 which an installation can modify directly. These files are located in the third tape file on the installation tape, and in Appendix B, “Sample Files” on page 83. The convention used for naming them is to give them a file name indicating the type of sample and a filetype of AP2MSAMP.

For example, IBM provides a sample for the LIBTAB APL2 file that cross-references library numbers to minidisks. As read from the tape, the file has the name LIBTAB AP2MSAMP.

Some of the files for which samples are provided are required for the normal operation of APL2; other files are optional. If the file is required and the I5688228 EXEC or I5688229 EXEC cannot find it on the disk being used to install APL2 during the generation of APL2, the sample file is copied and given the actual name required for use by APL2. For example, if the I5688228 EXEC or I5688229 EXEC cannot locate LIBTAB APL2, LIBTAB AP2MSAMP is copied to LIBTAB APL2. Note that the LIBTAB AP2MSAMP file remains unmodified, as supplied by IBM.

You should follow the same practice when updating any of the sample files that IBM supplies. Copy the sample file to the required name before editing it so you always have the original source for reference or backup.

Customizing Your APL2 Installation Process

This section discusses how to customize your APL2 installation process.

NAMES Files and the AP2M EXECs

The entire APL2 installation and maintenance structure uses the NAMEFIND command to search for tags in NAMES files. Essentially, NAMES files are used to contain tags, much like SCRIPT/GML tags, which describe a single logical entry. Each tag begins with a colon and terminates with a period. Any text following the period and preceding a subsequent colon is the value of the tag. Each logical entry is called a “nickname” and is designated by the :NICK. tag. NAMEFIND and NAMES files, described in *VM CMS Command and Macro Reference*, are the basis of the NOTE, SENDFILE, RECEIVE, and TELL commands.

APL2 NAMES Files

The AP2M EXECs, described below, search for the :NICK. tags of specific values to find instructions on how APL2 is to be customized. The EXECs are designed to search through the following two NAMES files in the order listed below when attempting to locate a given :NICK. entry.

© Copyright IBM Corp. 1984, 1994

1. AP2LOCAL NAMES

   Your local file contains your customizing information. A sample is provided on file 3 of the installation tape, under the name AP2LOCAL AP2MSAMP. If you want to use the sample file, copy it to the name AP2LOCAL NAMES and make changes in the copy.

   Using AP2LOCAL NAMES, you can customize the APL2 load modules to:

   - Control which parts of the product are loaded from tape, and where they are placed.
   - Exclude from the installation certain APL2 components whose supporting products are not installed.
   - Change or add information about saved segments to be used for APL2.
   - Add information about saved segments for GDDM and SQL/DS.
   - Add locally-written routines to the system.

   You can customize APL2 during your initial installation or during a regeneration of the APL2 system using the tags that drive the installation process.

   If the :NICK. tag is not found in your AP2LOCAL NAMES file, or if you do not have one, the EXECs look for AP2M220 NAMES.

2. AP2M220 NAMES

   This file contains the base description of the APL2 system as distributed by IBM for Version 2 Release 2 Modification level 0 (220). It contains the commands that generate each component, the parts that compose each component, and the format of the distribution tape. Use it as a reference when adding :NICK. tags to an AP2LOCAL NAMES file. A listing of the AP2M220 NAMES file appears in Appendix F, “AP2M220 NAMES File” on page 135.

AP2M EXECs

Each of the APL2 maintenance EXECs begins with the prefix AP2M. The AP2M EXECs will respond to a question mark as the first operand, with a description of the syntax and operation of the EXEC. Most of the AP2M EXECs are driven, either directly or indirectly, by tags in the NAMES files. The general functions of each EXEC are listed below:

- AP2MAINT

  AP2MAINT generates an entire system or a named component. This EXEC is called by the installation EXEC I5688228, or I5688229, and by the preventive service procedures when maintenance is applied to the system as a whole. The operands of the EXEC are the major :NICK. tag names in AP2M220 NAMES.

- AP2MLNK

  This EXEC is called by AP2MAINT to GENMOD modules, LKED LOADLIBs, and SAVESYS or SAVESEG saved segments. It follows the :NICK. tag names, which list the parts of a specific component. The operands of the EXEC name the output component, its type, and the source parts that generate it.

- AP2MREN

  This EXEC is called by AP2MAINT to rename parts shipped with service whose names are different than their execution-time names (for example, national language files).

- AP2MWS

  AP2MWS is called by AP2MAINT to install maintenance on workspaces. This EXEC invokes APL2 and issues)*IN* and)*SAVE* system commands to create an updated APL2 workspace. The operands describe the source part, output workspace identification, and password.

The remaining AP2M EXECs are not driven by the NAMES files:

- AP2M3800

  This EXEC is called by AP2MAINT when 3800 fonts are to be installed. AP2M3800 takes either the GENIMAGE or IMAGELIB argument and runs the commands based on that argument. The operands of this EXEC are the function and its associated list of fonts. For GENIMAGE, the list of fonts distributed in packed format with APL2 is in the AP2F3800 EXEC. For IMAGELIB, the list of TEXT decks is in AP2F3800 CNTRL.

  Privilege class A, B, or C is required to run IMAGELIB.

- AP2MCOPY

  This EXEC is invoked by you to copy execution-time files to production disks.

- AP2MV2AP

  This EXEC is invoked by you to combine VS APL auxiliary processor text decks with the APL2 auxiliary processor interface.

- AP2MP11L and AP2MP11M

  These execs are invoked by users to link-edit LOADLIBs or generate MODULEs from object files created by APL2 or non-APL compilers to make external routines accessible through processor 11.

NAMES Files Tags Processed by AP2M EXECs

When the AP2MAINT and AP2MLNK maintenance EXECs are operating, they attempt to read a :NICK. tag value from your installation's AP2LOCAL NAMES file. This permits you to alter the process of installation by altering or replacing individual parts or lists because AP2LOCAL is always searched before any NAMES files that IBM supplies. A description of the applicable tags and their use appears in Figure 5.

Figure 5 (Page 1 of 2). Syntax and Description of Installation Tags

Tag Syntax	Description
:NICK.*name*	Identify an entry. When a :NICK. entry of the desired value is found, it is searched, in order, for the following tags: :CMD., :LKED., :PART., :MSG., and :LIST..

Figure 5 (Page 2 of 2). Syntax and Description of Installation Tags

Tag Syntax	Description
:CMD.*command*	Specify an arbitrary EXEC2 statement or statements. Multiple statements are separated by a semicolon. If a nonzero return code results from any statement, the active :NICK. is abandoned immediately and any other tags within it are ignored.
:LKED.*card*	Specify an arbitrary string to be written following all input text decks to the linkage editor. For example: INCLUDE(ADMASP).
:PART.*filename filetype*	Specify the complete name of a file to be included in the build of the load module. If the *filetype* is omitted, a default of TEXT is assumed.
:MSG.*message*	Specify an arbitrary message to be typed at completion of current :NICK..
:LIST.*nick nick ...*	Specify a list of additional :NICK. tags to be processed.
:REM.*comment...*	Allow comments to appear in the names file.

Outline of Customizing Installation Steps

To customize your installation:

1. Execute the following CMS command:

   COPY AP2LOCAL AP2MSAMP A AP2LOCAL NAMES A

2. Edit the resulting file with a CMS editor such as XEDIT.

Once an AP2LOCAL NAMES file has been created, whenever the I5688228 EXEC, I5688229 EXEC, or AP2M EXECs are executed, any :NICK. in the AP2LOCAL NAMES will take precedence over the same :NICK. in the NAMES files that IBM supplies.

Installing to a Different Disk and/or Retrieving Optional Material

To avoid confusion between what IBM supplies with the base APL2 Licensed Program, the APAR fixes that have been applied, and the local modifying or customizing that has been done, your installation may prefer to use more than one minidisk to install and maintain APL2. Having the original files available on a separate disk when you are applying maintenance or customizing allows you to compare them to any changed files and easily isolate and correct problems that may occur as a result of the maintenance or customizing.

This section describes how you can control which files are loaded from the APL2 installation tape, and where they are loaded. An example shows how to load the files to a disk accessed as filemode G. This section also describes how you can control which disk the I5688228 EXEC, or I5688229 EXEC, generates the APL2 system on. An example shows how to customize locally and how to generate the system on a disk accessed as filemode E. You can put any IBM APAR fixes on a third disk and access it as filemode F. Because of the CMS search order, any file you supply locally will supersede any APAR fixes on the same file. Also, any

APAR fixes for a file supersede the original version of the file; but all versions will still be available if they are needed.

To customize your installation as described above:

1. Copy the first two tape files to a disk other than your A-disk after they have been loaded, or load them initially from the installation tape to the other disk.

   To install the first two tape files on your G-disk enter:

   ```
   VMFPLC2 LOAD * * G (EOF 2
   ```

2. Create a file named AP2LOCAL NAMES. Insert in it a line as follows:

   For the I5688228 EXEC, insert:

   ```
   :NICK.I5688228      :VMFPLC2-FMs.xxxxxxxxx
   ```

   For the I5688229 EXEC, insert:

   ```
   :NICK.I5688229      :VMFPLC2-FMs.xxxxxxxxx
   ```

   An example is shown in “AP2LOCAL NAMES Sample — AP2LOCAL AP2MSAMP” on page 85.

   Each letter in the value of the :VMFPLC2-FMs tag corresponds to the filemode for one of the nine tape files containing the APL2 product. If the value has an underbar in one of the positions, the corresponding tape file is skipped. Because the I5688228 or I5688229 EXEC does not reload the first two tape files, the first two letters in the tag :VMFPLC2-FMs serve only as place holders; the remaining letters align with the files to be loaded. With the default value of 'AAAAAA___', the I5688228 or I5688229 EXEC loads the next four files to the A-disk, and skips the last three (containing optional TEXT, VS APL migration aids for auxiliary processors, and 3800 fonts).

   Tell the I5688228 or I5688229 EXEC onto which disk to load the remaining tape files by specifying the appropriate file modes in the value of the :VMFPLC2-FMs. tag in the line entered.

   The I56882289 or I5688229 EXEC assumes that you have already loaded the first two tape files to disk. The EXEC checks to make sure that the tape is still positioned at the third tape file. It does not reload the first two tape files.

   To retrieve files three through six and also the VS APL migration aids file to the G-disk, change the value of :VMFPLC2-FMs to 'AAGGGG_G_'.

3. To put the files that the AP2M EXECs generate while APL2 is being installed and customized on a disk other than your A-disk, insert a line into the created AP2LOCAL NAMES file as follows:

   ```
   :NICK.PROLOG-AP2MAINT   :CMD.&-FM=LL
   ```

   where *LL* specifies the appropriate file mode.

   To put the files generated onto the E-disk, change the tag to :CMD.&-FM=E.

4. If you are performing a default installation, execute I5688228 or I5688229 INSTALL. Because of the tags in the created AP2LOCAL NAMES file, the tape files will be loaded and the APL2 system will be generated onto the specified disks, rather than on your A-disk.

5. If you are performing a customized installation, execute I5688228 or I5688229 LOAD. Because of the tags in the created AP2LOCAL NAMES file, the tape files will be loaded onto the specified disks, rather than your A-disk. Next,

incorporate the entries in the created AP2LOCAL NAMES file into the file copied from AP2LOCAL AP2MSAMP. Then execute I5688228 or I5688229 GENERATE. Again, because of the tags in the AP2LOCAL NAMES file, the APL2 system will be generated onto the specified disks, rather than on your A-disk.

The LIBTAB supplied assumes that you have installed on your A-disk so you will have to update it.

From now on, if you make all your local modifications on your E-disk, whenever you rerun I5688228 or I5688229 GENERATE it will pick up your local modifications first, build a new system, and place it on your E-disk. The files as originally supplied by IBM will remain isolated on the G-disk, and will always be available for reference. To make sure that the files on the G-disk are not altered, access the disk only in read-mode.

Removing GDDM

If you do not have or want GDDM, you can save space in the saved segment and LOADLIBs by removing related APL2 support.

In :NICK.PROLOG-AP2MAINT, change the tag :LIST.

To remove GDDM, the appropriate value is:

```
NOGDDM
```

The next time that you generate your APL2 system, the GDDM-related support is removed. You receive warnings about unresolved external references from the link-edits, but these can be ignored.

Removing SQL/DS

If you do not have or want SQL/DS, you can save space in the saved segment and LOADLIBs by removing related APL2 support.

In :NICK.PROLOG-AP2MAINT, change the tag :LIST.

To remove SQL/DS, the appropriate value is:

```
NOSQL
```

The next time that you generate your APL2 system, the SQL/DS-related support is removed. You receive warnings about unresolved external references from the link-edits, but these can be ignored.

Adding 3800 Font Support

Note: This section applies only to line-mode fonts. It is not required for APA fonts.

Because of the large amount of space required for 3800 font files, they are not loaded by default. If you want them, in :NICK.PROLOG-AP2MAINT, change the tag :LIST.

To add the 3800 fonts, the appropriate value is:

F3800

The next time that you generate your APL2 system, the 3800 font support is added.

If 3800 fonts are to be generated, model 1 fonts are generated by default. If you want model 3 compatibility mode fonts:

- Change the :CMD. tag in :NICK.GEN3800 to specify AP2F3803 instead of AP2F3800
- Change the value of the CPNAME parameter in the NAME3800 statement in the DMKSNT ASSEMBLE file to AP2F3803
- Change, if necessary, the value of the IMAGE parameter in the RDEVICE statement defining the 3800 printer in the DMKRIO ASSEMBLE file to AP2F3803.

Testing APL2 Without Saving Segments

You can suppress the actual saving of the segments during the APL2 default or maintenance installation.

In :NICK.SEGS-scmd, change the value SAVESEGS to TESTSEGS.

The APL2 installation process loads the nucleus extensions, but does not issue the segment saves.

Customizing Saved Segment Information for APL2

The IBM default is to save one copy of each APL2 loadlib. If you want to have one copy of AP2HIGH above the 16M line and another below, or two copies at different addresses for different sets of users, add d-HIGH2 to the list in :NICK.APL2SEG and define the extra segment.

If you change the names of the APL2 saved segments from the defaults used by IBM, you must change the name in the tag for each segment name changed. Samples are provided in the AP2LOCAL AP2MSAMP file.

Before saving a segment, the APL2 installation process looks to see if a “Clean-” tag has been provided for that segment. If you want to automatically issue PURGE and DEFSEG commands before each segment save, you can add these commands to the “Clean-” tag for the segments. Samples are provided in the AP2LOCAL AP2MSAMP file. Note that the user ID that issues the segment saves must have sufficient authority to issue any commands you place in these tags.

Adding Saved Segment Information for GDDM and SQL/DS

As supplied by IBM, the Installation Verification Program does not try to verify the parts of the product that use GDDM and SQL/DS, because it cannot determine the names of the shared segments for these products.

If you supply the shared segment names, the IVP does the added verifications. To supply them, add the segment names to the :LIST. tag in :NICK.GDDMSEGS and :NICK.SQLSEGS. If these products are installed, but not in shared segments, specify “NONE” as the only segment name.

Including a Modified APLSCOPT TEXT Deck

Note: This does not apply to APL2 Application Environment.

In :NICK.Local-SCOPT, duplicate the tags from :NICK.Local-OPT, then replace the filename AP2VIOPT with APLSCOPT in the tags. The next time you generate your APL2 system, your APLSCOPT TEXT deck will be included. APLSCOPT is required by)*MCOPY* to access VS APL workspaces. The default APLSCOPT from VS APL Release 4 is included. If you had both VS APL and modified APLSCOPT, you should include your modified APLSCOPT with APL2.

Making User-Written Auxiliary Processors Part of the System

A processor can be packaged in one of two ways:

- It can be resident with APL2 by link-edit and always available. This is preferred for production use because the code can be shared as long as the processor is reentrant.
- It can be retrieved from a LOADLIB at APL2 invocation with the APNAMES parameter. This is done while a processor is being developed and tested.

Creating a User-Written Auxiliary Processor

APL2 supports two types of auxiliary processors:

- Those written using the interface defined in *APL2/370 Programming: Processor Interface Reference* This interface should be used for all newly written auxiliary processors. Assembler mapping macros are provided for control blocks used in the interface, and these can be found in AP2MAC MACLIB, distributed within file 3 of the installation tape.
- Processors carried forward from VS APL, and so written as defined by *VS APL for CMS and TSO: Writing Auxiliary Processors*. The required VS APL service call and mapping macros are located in AP2VSAPL MACLIB, distributed within file 8 of the installation tape.

Conventionally, the numbers 100 through 499 are reserved to identify processors supplied by IBM, while numbers greater than 1000 are reserved to identify APL2 users to the global SVP. Therefore, the numbers 500 through 999 are normally used by installations to identify their auxiliary processors when signing on to the shared variable processor.

Testing a User-Written Auxiliary Processor

To make a user-written auxiliary processor available to APL2 for testing, you must:

1. Retrieve the appropriate macro library from the installation tape.
2. Place the processor into a LOADLIB.

   Two execs are provided for link-editing user-written auxiliary processors into LOADLIBs.

   The AP2MV1AP EXEC takes a version 1 (VSAPL) auxiliary processor TEXT deck and combines it with an APL2 interface module using the CMS LKED command. The result is placed in a LOADLIB you specify to the EXEC.

   The AP2MV2AP EXEC takes a version 2 (APL2) auxiliary processor TEXT deck and combines it with an APL2 interface module using the CMS LKED command. The result is placed in a LOADLIB you specify to the EXEC.

The syntax of the AP2MV2AP EXEC (and the AP2MV1AP) EXEC is:

AP2MV2AP *Entry* TEXT*id* LOADLIB*id* (LKED*options*

where:

Entry Processor entry point name

TEXT*id* The filename filetype filemode of TEXT deck

If an auxiliary processor is made up of multiple text decks, use the CMS COPY command with the APPEND option to place all the text decks into a single file and specify that filename, filetype, and filemode.

LOADLIB*id* The filename filetype filemode of LOADLIB for output

LKED*options* Optional LKED command options

For example, to make an AP out of AP517 TEXT with an entry point AP517 and the result going in library MYAP:

EXEC AP2MV2AP AP517 AP517 TEXT * MYAP LOADLIB A

3. Issue the FILEDEF command for the LOADLIB.

   If your processor is in load library MYAP, you can make it available to the APL2 session with the following CMS command:

   FILEDEF AP2LOAD DISK MYAP LOADLIB *

   This command can conveniently be placed in your AP2EXIT EXEC.

   Multiple LOADLIBs may not be concatenated. That is, if you use the CONCAT option of the FILEDEF command, APL2 is unable to process the LOADLIBs. If you have multiple APs, you should specify the same LOADLIB on the AP2MV2AP EXEC or you should use the CMS LOADLIB command to copy the load modules into a single LOADLIB.

4. Invoke APL2 with the APNAMES invocation parameter.

   Include AP(AP517) on your APL2 invocation command to tell APL2 to start your processor.

Making a User-Written Auxiliary Processor Resident

A resident auxiliary processor is one that is link-edited as part of the APL2 module. The names of user-written auxiliary processors and any additional entry points must not conflict with external names in the APL2 link-edit map.

Auxiliary processors that are automatically available are called resident auxiliary processors and must be specified in the RESAPS installation parameter. The RESAPS installation parameter is a parameter of the AP2VOPT macro in the AP2VIOPT or AP2VIOPE module. For more information about the RESAPS parameter, see “Changing the APL2 System and Invocation Options” on page 62. In addition to being specified in RESAPS, the TEXT deck for the auxiliary processor must be included in the link-edit of the APL2 LOADLIB performed by the AP2MAINT EXEC during installation and maintenance. To include the TEXT deck in the link-edit, you must modify the Local-TEXT entry in the AP2LOCAL NAMES file.

For example, to add a processor named AP517 to your system, if the current Local-TEXT entry is:

```
:NICK.Local-TEXT :LIST.Local-OPT Local-SCOPT
```

alter the current entry to:

```
:NICK.Local-TEXT :LIST.Local-OPT Local-SCOPT AP517
```

and add an entry:

```
:NICK.AP517      :PART.AP517 TEXT
```

The next time you generate your APL2 system, the AP is added. The addition of user-written auxiliary processors increases the virtual storage requirements of APL2. Therefore, you should check to see that the new load module still fits within the saved segment for APL2.

Making User-Written External Routines Part of the System

Associated Processor 11 provides facilities through which external routines written in FORTRAN, C/370, PL/I, Assembler Language, or APL2 may be executed. Processor 11 provides facilities with which you can specify how APL data should be mapped to and from the data structures used by these programs.

APL2 routines must be processed using the PACKAGE external routine supplied with APL2 (described in *APL2/370 Programming: System Services Reference*). The result of using PACKAGE is a reentrant, refreshable TEXT deck on your A-disk, ready to be packaged.

External routines may be packaged for APL2 in a number of ways:

- Link-edited into LOADLIBs
- Link-edited into LOADLIBs and placed in saved segments
- Link-edited with APL2
- Made CMS nucleus extensions
- Placed in TXTLIBs
- Accessed as TEXT decks.
- GENMODed into modules

External routines link-edited into LOADLIBs do not require storage when not in use, and when in use can be relocated by APL2 wherever storage is available in each user's virtual machine. Each user must have their own copy in their own virtual machine.

External routines link-edited with APL2 do require storage when not in use, and must be reentrant and refreshable, but if APL2 is installed in a saved segment, a single copy of the routine can be shared by all users.

External routines that exist as nucleus extensions may have various properties and requirements since nucleus extensions may be generated in various ways, and it is up to you or your users to generate the nucleus extensions. Except for external routines link-edited with APL2, APL2 does not generate the nucleus extensions for you. You must do that yourself, but you are free to do it in any way you find possible. One common way is to use the CMS NUCXLOAD command.

A complete description of the procedures for preparing non-APL routines and APL2 workspaces for use through processor 11 can be found in *APL2/370 Programming:*

System Services Reference. Those discussions also include details on link-editing APL2 namespaces and non-APL routines and processor 11's search order rules.

Installing Namespaces and Non-APL Routines in VM Saved Segments

APL2 namespaces and non-APL routines to be accessed through $\square NA$ can be placed in saved segments. Once placed in a saved segment, the namespaces and routines can be shared between multiple APL2 users. These procedures do not require the namespaces or non-APL routines to be link-edited with APL2; further, they can be modified, updated and re-saved without having to reinstall APL2.

Routines written in languages other than APL and accessed through $\square NA$, can be placed in a saved segment and shared between users if they are reentrant (note that Fortran programs are not normally reentrant).

Namespaces and non-APL routines are placed in a saved segment by first link-editing them into a member of a load library and then placing that member in a saved segment. There are several ways namespaces and non-APL routines can be combined into a member. However, processor 11's support varies depending on which way is used to combine namespaces and non-APL routines.

1. Loading one member containing one or more namespaces into a saved segment.
2. Loading one member containing one or more non-APL routines into a saved segment.
3. Loading one member containing a combination of one or more namespaces and non-APL routines into a saved segment.
4. Loading multiple members each containing one namespace and optionally one or more non-APL routines into multiple overlapping saved segments.

   Overlapping saved segments may be used even if the members in them are required simultaneously by the APL user. Processor 11 automatically handles this for the user and ensures proper access to the correct overlapping namespace or routine. Overlapping saved segments does, however, result in a small performance degradation.

   When members in overlapping saved segments are used, each member must contain one (and only one) namespace. Each member may also contain non-APL routines; they may be self-describing or described by NAMES file entries. If a member contains non-APL routines, it must also contain a routine list. No member in an overlapped saved segment can contain only non-APL routines (and no namespace.)

Placing multiple members into overlapping saved segments requires the least storage since the saved segment storage is reused for each member. The other techniques require more storage since multiple nonoverlapping saved segments are required to load multiple members.

Members that contain more than one namespace or non-APL routine must also contain a routine list that informs processor 11 where to find the namespaces and routines within the member.

The procedure for creating namespaces is described in the discussion of the $PACKAGE$ function in *APL2/370 Programming: System Services Reference*.

The procedure for writing and preparing non-APL routines for use from APL2 is described in *APL2/370 Programming: System Services Reference*.

The procedures for creating a routine list and link-editing namespace and non-APL routine object files into a member of load library are also described in *APL2/370 Programming: System Services Reference*. Use the LIST XREF MAP options on the LKEDIT command. This produces an output file showing the size of the member in hex bytes.

Placing a Member in a Saved Segment

Once a member has been built containing namespaces or non-APL routines, it can be placed in a saved segment. The following procedure can be used to place a member in a saved segment. Note that the procedure must be performed multiple times if multiple segments are to be used, whether or not they overlap.

1. Leaving some room for expansion, ask your systems programmer to define a saved segment for you of a size indicated in the output file of the link-edit process. If several members are to be overlapped, choose a size that will hold the largest member. Negotiate a meaningful name for the saved segment with the systems programmer.

   Note: It is not possible to overlap namespace and external routine saved segments with the APL2 saved segment. You can only overlap them with other namespace and external routine saved segments. Further, each of the overlapped saved segments must begin at the same address. For example:

   Valid saved segment maps:
   Namespace 1 at 7F0000
   Namespace 2 at 7F0000
   Namespace 3 at 7F0000
   Namespace 4 at 6F0000 (As long as it doesn't overlap
   Namespace 5 at 6F0000 the segment starting at 7F0000)

   Invalid saved segment maps:
   Namespace 1 at 7F0000 (Assume it's more than X'1000' bytes)
   Namespace 2 at 7F1000 (Invalid because it partially overlaps
   Namespace 1)

2. Log on to a VM ID with the privilege class to issue a SAVESYS (SP) or SAVESEG (XA) command, by default CLASS E.
3. Issue a filedef for the load library containing the link-edited member to be loaded.

   `FILEDEF` *ddname* `DISK` *name* `LOADLIB *`

   where *ddname* is any valid ddname and *name* is the name of the LOADLIB file
4. Issue an AP2XLOAD command to cause the member from the load library to be placed in the saved segment. The format of the AP2XLOAD command is:

   `AP2XLOAD` *nucname member ddname* `(SEGMENT` *segment*

   where *nucname* is anything you want, *member* is the name of the member in the load library, *ddname* is the ddname to which we FILEDEFed the load library and *segment* is the name of the saved segment defined in step 1 above.

5. To ensure proper storage recovery, clear the filedef.

   ```
   FILEDEF ddname CLEAR
   ```

   This releases the OS storage required for the filedef, making that storage available for other use and reducing the potential for storage fragmentation.

   The LOADLIB file is no longer needed in the process. It can be discarded.

6. Use the CMS SETKEY command to set the storage key to 0.

   ```
   SETKEY 0 segment
   ```

   The current setting of the storage key is associated with the segment when it is saved and will to some extent preclude other programs from modifying the information in the segment. Since user programs and disk-resident CMS commands generally execute with a PSW key of X'E' (decimal 14) this provides a level of protection for the segment from inadvertent destruction.

7. Cause the saved segment, loaded in step 4 on page 57, to be saved.

   Use the SAVESYS command:

   ```
   SAVESYS segment
   ```

   or the SAVESEG command:

   ```
   SAVESEG segment
   ```

   as appropriate for your version of VM.

8. Reset the storage key to the normal user value, 14.

   ```
   SETKEY 14 segment
   ```

   That completes the creation of the saved segment and saving the member into it. Now the application(s) in the saved segment are available to all APL2 users for use.

   Note: It is also possible to load a saved segment from a MODULE file rather than from a member of a load library. Because the intermediate file is only temporary anyway, only one technique is shown here. If loading from a MODULE file is attempted, be sure to use the `RLDSAVE` option on the `LOAD` command to save the relocation information.

Accessing a Saved Segment Containing a Namespace

Each APL2 user who wants to access an object in a saved segment which contains a namespace must:

1. Issue the following CMS command:

   ```
   AP2VUTIL SEGMENT LOADPW segment nucname
   ```

where *segment* is the name of the saved segment defined by your systems programmer, and *nucname* is the CMS nucleus extension name the user wants to assign to the saved segment. If the segment contains one package, *segment* and *nucname* can be the same or different. If the saved segment contains multiple members, then the command must be issued for each member in the saved segment, and *nucname* should match the name of each member. For example:

```
AP2VUTIL SEGMENT LOADPW segment nucname1
AP2VUTIL SEGMENT LOADPW segment nucname2
....
AP2VUTIL SEGMENT LOADPW segment nucnamen
```

2. Under APL2, issue □*NA*'s for objects in the saved segment.

   Once a nucleus extension name has been assigned, APL workspace names can be associated with namespace objects or non-APL routines in the saved segment. The member name specified in the left argument of □*NA* or the :MEMB. tag in the NAMES file should match *nucname*. For example:

   ```
   'nucname' 11 □NA 'OBJECT'
   ```

3. When use of the saved segment is complete and all □*NA* links to it are broken (by □*EX*,)*CLEAR*,)*LOAD*, or)*OFF*), issue the following CMS command to delete the CMS nucleus extension and release the saved segment:

   ```
   AP2VUTIL SEGMENT PURGEPW segment nucname
   ```

   This command should be issued for each *nucname* that was loaded. This step is not required, but is good practice.

This procedure should be used when the saved segment contains one or more namespaces. The procedure can be used if the saved segment also contains one or more non-APL routines.

Note: The AP2VUTIL command mentioned in steps 1 on page 58 and 3 above can be issued:

- outside APL2, for example in the EXEC that invokes APL2
- inside APL2 using the)HOST command or AP 100.

Accessing a Saved Segment Containing only Non-APL Routines

Each APL2 user who wants to access an object in a saved segment which contains only non-APL routines must issue the following command:

```
AP2VUTIL SEGMENT LOADA segment nucname
```

where *segment* is the name of the saved segment in which the non-APL routine resides, and *nucname* is the CMS nucleus extension name the user wants to assign to the saved segment.

Once the nucleus extension name has been assigned, APL workspace names can be associated with routines in the saved segment. The member name specified in the left argument of □*NA* or the :MEMB. tag in the NAMES file should match *nucname*. For example:

```
'nucname' 11 □NA 'OBJECT'
```

When use of the saved segment is complete and all ⎕*NA* associations to it are broken (by ⎕*EX*,)*CLEAR*,)*LOAD*, or)*OFF*), issue the following CMS command to delete the CMS nucleus extension and release the saved segment:

```
AP2VUTIL SEGMENT PURGEPW segment nucname
```

Link-Editing External Routines with APL2

External Routines can be link-edited directly into the APL2 LOADLIB, and into the APL2 saved segment if one exists. If an APL2 saved segment does exist, then a single copy of the external routine will be shared between all users of the saved segment.

Routines to be link-edited into the APL2 saved segment must be reentrant and refreshable. A routine that is reentrant and refreshable does not modify itself. TEXT decks produced by the PACKAGE external routine are reentrant and refreshable. Assembler language routines may or may not be reentrant and refreshable depending on how they were written. FORTRAN routines compiled with VS FORTRAN are not reentrant and refreshable (see *VS FORTRAN Programming Guide* for information on how to make portions of a FORTRAN program sharable by using the VS FORTRAN separation tool).

To link-edit an external routine with APL2:

1. Edit AP2VIOPT ASSEMBLE or AP2VIOPE ASSEMBLE and add the entry name of your external routine to the RESEPS parameter.

   If the non-APL routine is self-describing, add the name of the routine description, not the name of the routine itself, to the RESEPS parameter. See “Changing the APL2 System and Invocation Options” on page 62.

2. Create an AP2LOCAL NAMES file if you don't already have one by copying AP2LOCAL AP2MSAMP and supplying the NAMES filetype.

3. Edit your AP2LOCAL NAMES file so that the Local-EPs :NICK. includes a :LIST. NICK identifying your local external routines, and then add an additional :NICK. identifying the TEXT deck for each routine.

   For example, to add the routine in the ROUTINEA TEXT deck, change the Local-EPs :NICK. to:

   ```
   :NICK.Local-EPs :LIST. ROUTINEA
   ```

   and add a :NICK.:

   ```
   :NICK.ROUTINEA  :PART. ROUTINEA TEXT
   ```

   If the non-APL routine is self-describing, add :NICK. tags identifying the names of the routine description and add the names of the :NICK. tags to the :LIST. tag included in the Local-EPs :NICK. tag.

4. If the non-APL routine is self-describing, simply use the name of the routine description as the member name in the left argument of ⎕*NA*. If the non-APL routine is not self-describing, describe the routine in a NAMES file using a :MEMB. tag to identify the text deck entry point. To prevent processor 11 from trying to load the routine from a LOADLIB, do not code a load library name in the left argument of ⎕*NA* nor code the :LOAD. tag in the NAMES file.

Customizing the Execution of APL2

This section discusses how to customize the execution of APL2.

Customizing the APL2 Session Manager Profile

Note: This does not apply to APL2 Application Environment.

When used, the APL2 session manager automatically executes a PROFILE LOAD command as part of APL2 invocation. This profile establishes the default values for session manager parameters such as program function keys, log size, and display attributes. See “DEFAULT VSAPLPR Sample — DEFAULT AP2MSAMP” on page 113 for more information. Individual users may modify the default session manager profile to meet their own requirements or they can create additional profiles.

The file identification is:

```
DEFAULT VSAPLPR *
```

The filemode (*) identifies the APL2 system disk.

IBM recommends that you use a terminal with the APL feature to edit the session manager profile. Most named editors delete all trailing blanks from variable-length record files. If trailing blanks are required on some program function key definitions, those definitions should only be changed under control of the APL2 session manager.

For a description of the APL2 session manager and the session manager profile, see *APL2/370 Programming: System Services Reference*.

Changing the APL2 System and Invocation Options

System Options

You can tailor certain aspects of APL2 operation for your particular system or location. Update the system options when you want to:

- Change the saved segment names used when referencing APL2
- Change the user ID of the global SVP service machine
- Make your own auxiliary processors resident with APL2
- Change default invocation options.

Invocation Options

The command to invoke APL2 allows you to specify several options that affect your APL2 session. These options include:

- The characteristics of the terminal you are using
- The size of your workspace and the size of storage areas used with auxiliary processors
- Debugging options to be in effect.

The invocation options are summarized alphabetically by keyword in Figure 3 on page 27 and described in detail in Chapter 4, “APL2 Invocation and Termination” on page 25.

Changing the Options

To change the APL2 system options:

1. For APL2, copy the AP2VIOPT AP2MSAMP file to AP2VIOPT ASSEMBLE. For APL2 Application Environment, copy the AP2VIOPE AP2MSAMP file to AP2VIOPE ASSEMBLE.
2. Edit AP2VIOPT, or AP2VIOPE, ASSEMBLE.

   Modify the operands of the AP2VOPT macro instruction and change the values for the system option parameters to meet your installation requirements. APL2 system options and their default values are summarized in Appendix D, “APL2 System Options” on page 129. The format of the macro instruction is:

   AP2VOPT *parm1*=*val1*, *parm2*=*val2*,...*parmn*=*valn*

   parm1 through *parmn* The installation parameters specified in any order

   val1 through *valn* The values you can specify for the parameters

   For example, to make the APL2 messages appear in French, change the value for the QNLT parm to ‘FRANCAIS’.

   AP2VOPT QNLT=FRANCAIS,AP2SHNM=(LCLAPL1,LCLAPL2)

   In modifying the default invocation options, you should be aware that certain auxiliary processors are required for the correct execution of certain system commands; that is, the copy commands)*COPY*,)*PCOPY*, and)*MCOPY* require AP 104, the)*HOST* command requires AP 100, and the)*IN* and)*OUT* commands require AP 110. Therefore, do not change the system or invocation options in such a way that these auxiliary processors are not available.

After making the appropriate changes to the file AP2VIOPT ASSEMBLE, or AP2VIOPE ASSEMBLE, then:

3. Issue GLOBAL MACLIB AP2MAC to make the APL2 macros available.

4. Issue ASSEMBLE AP2VIOPT, or ASSEMBLE AP2VIOPE, to generate the TEXT deck.

The next time you regenerate your APL2 system, the new AP2VIOPT TEXT or AP2VIOPE TEXT will be used.

Changing the APL2PI Command Options

You can tailor how APL2PI invokes APL2 for your system or location. Update the command options when you want to:

- Change the name of the APL2 module that should be used when invoking APL2 from APL2PI.
- Specify that the APL2 Application Environment module should be used when invoking APL2 from APL2PI.
- Specify the invocation options that should be used by APL2PI.

Changing the Options

To change the APL2PI command options:

1. Copy the AP2XAPIC AP2MSAMP file to AP2XAPIC ASSEMBLE.
2. Edit AP2XAPIC ASSEMBLE.

   Following the guidelines in the module's comments, modify the values of the command and options to meet your system requirements.
3. Issue GLOBAL MACLIB AP2MAC to make the APL2 macros available.
4. Issue ASSEMBLE AP2XAPIC to generate the TEXT deck.

The next time you regenerate your APL2 system, the new AP2XAPIC TEXT will be used.

Using Nondefault Saved Segment Names for APL2

For APL2: Follow the procedure in “Changing the Options” on page 62 and update the AP2SHNM parameter in AP2VOPT macro in AP2VIOPT module. Reassemble the module. Then create or modify an AP2LOCAL NAMES file with the :NICK. tags “d-SEG1,” “d-SEG2,” “d-HIGH1,” and “d-HIGH2” copied from AP2M220 NAMES. The saved segment names (AP2V22V1, AP2V22V2, AP2V22W1, and AP2V22W2, respectively) in these :NICK. tags should be changed to match what you have named the APL2 shared segments. You should then run AP2MAINT APL2SEG to resave the saved segment. Run AP2MAINT M-APL2 to regenerate the stub module that knows which saved segment to load.

For APL2 Application Environment: Follow the procedure in “Changing the Options” on page 62 and update the AP2SHNM parameter in AP2VOPT macro in AP2VIOPE module. Reassemble the module. Then create or modify an AP2LOCAL NAMES file with the :NICK. tags “d-AESEG1,” “d-AESEG2,” “d-AEHIGH1,” and “d-AEHIGH2” copied from AP2M220 NAMES. The saved segment names (AP2V22E1, AP2V22E2, AP2V22F1, and AP2V22F2, respectively) in these :NICK tags should be changed to match what you have named the APL2 shared segments. You should then run AP2MAINT APL2AESEG to resave the saved segment. Run AP2MAINT M-APL2AE to regenerate the stub module that knows which saved segment to load.

Modifying the AP2EXIT EXEC

The AP2EXIT EXEC is invoked when APL2 is initialized and when it terminates. The EXEC is normally used for issuing commands to redefine your virtual machine, or for logging APL2 usage to a service machine. You can modify the AP2EXIT EXEC to conform to your own installation standards; to do this see “AP2EXIT EXEC Sample — AP2EXIT AP2MSAMP” on page 83.

For example, if GDDM is not installed in a saved segment and you want to use GDDM TXTLIBs, you could modify the following statement in the AP2EXIT EXEC so that the GDDM code is available:

```
*GLOBAL TXTLIB ADMPLIB ADMGLIB
```

To modify the statement:

1. Remove the `*`.
2. If the GDDM TEXTLIBs haven't been merged into ADMGPLIB, change the name of the GDDM TXTLIB from the default, ADMGPLIB, to the name or names of the GDDM TXTLIBs on your system.

In the case of a nonshared GDDM, when invoking APL2 you should specify a value of at least 500K-bytes for FREESIZE in the invocation options.

Defining APL2 Libraries

APL2 libraries are used to map a given library number and workspace name or APL data files file name to a specific CMS file name and filetype. The file name and filetype are further identified with a single filemode letter or with a single user ID and minidisk address.

Note: APL2 Application Environment uses libraries only for APL data files. Workspaces are not distributed with APL2 Application Environment.

The distinction between public and private libraries in APL2 is a matter of convenience and practice rather than one of rigid definition and required use. Just like CMS, where the Y-disk traditionally holds shared read-only files and the A-disk is a private work area, the APL2 library numbers have many conventions, but few rules.

What an APL2 Library Number Means

An APL2 library number refers to a single filemode letter or to a single user ID and minidisk address. Several library numbers may refer to the same disk. For the CMS file system, the only real distinction is in the file type used. Private workspaces have the fixed file type of APLWSV2; public ones have V*nnnnnnn*, where *nnnnnnn* is the library number padded with leading zeros. For example, public library workspace 1 WSINFO is a CMS file with a file name of WSINFO and a file type of V0000001. The practical consequences are that several public library numbers can share the same minidisk without conflict. However, if several private library numbers refer to the same minidisk, then a private workspace on that minidisk can be accessed through any of the private library numbers, giving the appearance of multiple (distinct) workspaces when there is only one.

Since workspaces in PRIVATE libraries have the same filetype, it's dangerous to assign two PRIVATE libraries to one disk. Suppose PRIVATE libraries 1003 and 1004 point to the same disk and the disk contains the PRIVATE workspace F00. Since)LIB shows that you have a workspace 1003 F00 and a workspace 1004

F00, you may mistakenly think you have two workspaces. If you)*DROP* 1003 F00 however, you will discover that 1004 F00 is gone too. If you want to define multiple libraries on one disk, define at most one PRIVATE library, and use PUBLIC libraries for the rest.

The minidisks for both public and private libraries may be referred to by either defining the filemode at which they are found, or by specifying the user ID and disk address to which APL2 should dynamically link. Specify a file mode if you normally have the disk accessed outside of APL2 anyway, or if you use the library frequently and other users will not normally be updating the disk. Access the disk in your AP2EXIT EXEC if you don't have the disk accessed outside of APL2. Dynamically link if other users will want to update the disk while you are running APL2, or if you don't normally have the disk accessed outside of APL2 and you seldom use it in APL2.

When a library is defined for dynamic linking, APL2 connects to the minidisk only while a file is actively in use. For workspaces, this is limited to the time it takes to perform a library system command. Since no single user has write access to the minidisk for long, it is possible for several users to cooperate in sharing the disk space for a common set of workspace libraries. Of course, this technique is less useful for APL data files because they are often kept open for longer periods of time.

When a library is defined at a specified filemode, APL2 assumes that the disk is permanently accessed and tries to read or write it immediately upon receiving a request from the APL2 session. This is clearly more efficient for personal mini-disks, such as your A-disk, or for read-only libraries.

Public Libraries: The numbers 1 to 999 are traditionally used for public libraries. These libraries contain APL2 programs and utilities of common interest. The work-spaces IBM supplies with APL2 provide examples of APL2 usage, programs that access GDDM, VSAM, and SQL/DS from APL, and the installation verification program.

During installation, the workspaces supplied with APL2 are loaded with filetypes consistent for public libraries 1 and 2. Approximately 2000 1K-byte blocks are required for the workspaces supplied with APL2.

Private Libraries: Traditionally, libraries with numbers above 1000 have been used for private libraries or for libraries associated with groups of users engaged in a project. The private library is frequently written by a single user, although it may be read by others.

Space allocation for libraries is dependent on virtual machine size, the value selected for WSSIZE, your requirement for APL data files, and the number of work-spaces saved. An APL2 workspace can easily range from 20 to over 300 1K-byte blocks, depending on the application, the size of the virtual machine, and the value of the WSSIZE invocation option.

Determining Your Default Library

Each APL2 session has an identification number determined by you through an invocation option, or by the installation through default or assignment by the global SVP. Traditionally, the default number in CMS has been 1001. Unless the global SVP is in use, every APL2 session can use the same number because there is no communication between virtual machines. This account identification number is used during the APL2 session as the default library for workspaces. Normally you will want this library to be your A-disk; the sample IBM supplies makes this assignment.

Defining Libraries with LIBTAB APL2

The CMS file LIBTAB APL2 defines private and public APL2 libraries. A LIBTAB entry consists of:

```
TYPE LIBNO SOURCE
```

where:

TYPE	PUBLIC or PRIVATE
LIBNO	*libnum* or (*libnum1 libnum2*)
SOURCE	*owner device w-l-m* or *@ filemode*

The components of LIBTAB APL2 entries are:

PRIVATE or PUBLIC	The library type (only the first three letters are required).
libnum	The library number, which can be any integer from 0 to 9999999. If 0, the *owner device w-l-m* or *@ filemode* is used if the library number being referenced matches the first element of ⎕*AI*.
(libnum1 libnum2)	A range of library numbers, *libnum1* begins the range and *libnum2* ends the range.
owner device w-l-m	The user identification of the owner, the virtual device address of the private library disk in the owner's directory, and the mode in which the disk should be linked when APL2 wants write access to it. The “write-link-mode” (*w-l-m*) is optional. Enter W for write-access, or M for multi-access. If not specified, W will be used as the access mode.
@ filemode	Specifies that the library *libnum* is on the minidisk accessed as *filemode*. If the *filemode* is not currently accessed, the message LIBRARY NOT AVAILABLE is displayed.

To define new libraries, add entries to LIBTAB APL2.

- Entries must be entered in uppercase characters only.
- Entries must begin in column 1.
- Entries may not be longer than one line.
- Operands must be separated by one or more blanks.
- Blank entries or entries that begin with an asterisk are completely ignored.

At the conclusion of the AP2EXIT EXEC, APL2 users must have access to a copy of LIBTAB APL2 on an accessed disk, or else they will only have access to workspaces on their A-disk.

The file LIBTAB AP2MSAMP supplied with the APL2 Licensed Program is shown in “LIBTAB APL2 Sample — LIBTAB AP2MSAMP” on page 114. If you want to use the sample, you must copy it and rename it LIBTAB APL2.

Private libraries do not have to be defined during the customizing process because the private library 0 is already defined, as on the user's A-disk in the sample. The private library 0 entry is used whenever the library being referenced matches your default library.

The sample LIBTAB APL2 contains entries for private libraries 1001 through 1026, which correspond to the 26 possible CMS filemodes, so that you can conveniently refer to any accessed disk. If you determine that particular users require libraries in addition to the default, you can define additional private libraries for those users or you can let them copy your system LIBTAB APL2 and modify it themselves.

For example, to define a private library with a library number of 2003 for any APL2 user with access to minidisk G, add the entry:

```
PRIVATE 2003 @ G
```

To redefine public libraries 1 through 99 as being on the Y-disk rather than the A-disk, change:

```
PUBLIC  (1 99) @ A
```

to:

```
PUBLIC  (1 99) @ Y
```

To define a public library with a library number of 4002 owned by user APLUSER as the user's 191, add the entry:

```
PUBLIC 4002 APLUSER 191
```

To define public library 599 on your F-disk, add to LIBTAB APL2:

```
PUBLIC 599 @ F
```

Installation Exit for the Global SVP

When the global SVP is installed, each APL2 session invocation attempts to identify the user to the global SVP service machine. You may provide an exit, in the form of a CMS command named AP2SVPEX. (“Sample Exit for Global SVP — AP2SVPEX AP2MSAMP” on page 92 contains a sample exit.) If you do not supply the command, the SVP service machine will permit any user who specifies a nonzero ID number access to the global SVP.

Your AP2SVPEX command may be used to:

- Authorize or deny access to the global SVP for the current APL2 session. When authorized, the service machine reveals the name of the unprotected saved segment for use. Authorization is not a security feature. The users are assumed to be cooperating with each other. After the users know the name of the unprotected saved segment, they can inspect or modify any part of it.
- Provide or change your APL2 identification number. You may supply this number with the ID invocation option, but your command can select an ID for you.

Product-sensitive programming interface

The global SVP service machine attempts to execute the command AP2SVPEX IDENTIFY each time you invoke APL2. Your command should reside on a disk accessed by the global SVP service machine, such as its A-disk. The command is needed only by the global SVP, not by typical APL2 users.

When control is given to your command, register 1 points to the command AP2SVPEX IDENTIFY, and register 0 points to an extended parameter list for the same command. The EPLOPT field of the extended PLIST points to a VMCF external interrupt buffer, mapped by VMCMHDR in the VM copy code, VMCBLOK. All registers follow the convention for CMS commands. Your command should return with register 15 set to 0 to authorize access, or set to 1 to deny it.

Your command may also change the value of the processor ID by changing the VMCMMID field of the VMCMHDR parameter list. Upon entry, this field contains the ID value supplied when APL2 was invoked. Changing it to a value of 0 also has the effect of denying user access. On successful completion of this initialization, APL2 uses the value of VMCMMID, if it is nonzero, as the first element of □*AI* and as your default private library number.

The sample command shown in “Sample Exit for Global SVP — AP2SVPEX AP2MSAMP” on page 92 invokes an EXEC named SVPEXIT EXEC, which can indicate by its return code whether you are authorized. Restrictions that apply to the command and the EXEC are documented in the sample command. For details on writing CMS commands, register conventions, extended parameter lists, and the VMCMHDR control block, see *VM System Programmer's Guide*.

End of Product-sensitive programming interface

Preparing for Access of SQL/DS by APL2

Individual Users Who Want to Access SQL/DS Must:

1. LINK and ACCESS the minidisk in the SQL/DS machine.

   ```
   LINK SQLMAINT 195 195 RR
   ACC 195 Q
   ```

   The user ID may be different on your system.

2. Set the language to be used by SQL/DS for messages.

   ```
   SET LANGUAGE AMENG (ADD ARI USER
   ```

   This example shows American English.

3. Initialize your user machine to the database by issuing the `SQLINIT` EXEC (if not already done).

   ```
   SQLINIT DBNAME(SQLDBA) [PROTOCOL(DRDA)]
   ```

   The `SQLINIT` process places onto your A-disk some files that are the SQL/DS "bootstrap" modules. These modules determine which SQL/DS database is accessed. Once the files are created, there is no need to repeat this step unless they are erased or you wish to change to a different database.

The SQL/DS Database Administrator Must:

1. Be able to access SQL/DS (See above)
2. Preprocess the ASMSQL file supplied with APL2 using the SQL/DS SQLPREP EXEC.

   The purpose of this task is to identify the APL2 packages to any IBM relational databases that are accessed from APL2 on VM.

   The following are examples of the two SQLPREP steps required for APL2. SQL/DS Version 3 Release 3 or higher is required.

   ```
   SQLPREP ASM PP(NOPUNCH,NOPRINT,PREP=SQLDBA.AP2V2R2R,
           USER=SQLDBA/sqldbapw,ISOL(RR),ERROR,BLOCK)
           IN(AP2V127I ASMSQL fm)
           [DBLIST(dbname1, dbname2,...)]

   SQLPREP ASM PP(NOPUNCH,NOPRINT,PREP=SQLDBA.AP2V2R2C,
           USER=SQLDBA/sqldbapw,ISOL(CS),ERROR,BLOCK)
           IN(AP2V127I ASMSQL fm)
           [DBLIST(dbname1, dbname2,...)]
   ```

   The two SQLPREP commands should both be issued as shown, except that the BLOCK parameter is optional. Because the use of BLOCK provides significant performance improvements on VM, it is highly recommended. In addition, depending on the requirements of your installation and users, you may want to add the DATE, TIME, or CCSID parameters.

   The SQLPREP statements may be run against any database or list of databases accessible through the DRDA protocol. If APL2 is used only to access the local database on VM, the DBLIST parameter need not be used. If APL2 is used to access remote databases, the DBLIST parameter is used to specify the list of those databases, and the SQLPREP must be run with PROTOCOL(DRDA) (a parameter to SQLINIT) in effect.

   When running the SQLPREP statements against a VM SQL/DS system, you can expect a return code of 4. (The ERROR parameter to SQLPREP allows DB2-only statements in the APL2 file to be ignored when running against SQL/DS.)

   See *SQL/DS Application Programming for IBM VM Systems* for complete details on the SQLPREP command.
3. Grant authority for users to use the APL2 interface.

   ```
   CONNECT SQLDBA IDENTIFIED BY sqldbapw
   GRANT RUN ON AP2V2R2R TO PUBLIC
   GRANT RUN ON AP2V2R2C TO PUBLIC
   ```

   The CONNECT and GRANT commands can be issued using the ISQL processor, with the SQLDBSU EXEC, or from APL2 using AP 127.

If the installation has multiple SQL/DS databases, and APL2 is used from each of them, these steps must be repeated for each database.

Extending or Adding National Language Message and Command Files

A set of national language message and command keyword definition files has been distributed with APL2. These files may be tailored by an installation using standard editors. Each file is identified by a 1- to 8-character language name (called “lang” here) that can be assigned to the $\square NLT$ system variable.

Each file is named "lang APL2LANG", and may contain either fixed or variable length records. The file may be on any accessed minidisk; standard search order is used.

The syntax used for defining message and command text permits trailing comments. The record formats (defined below) permit trailing comments in any record. This also means that records may be numbered on the right, and any such numbers will be ignored by APL2, so long as column 72 is blank. If the records are not numbered, they may contain data through column 80. No data in the records is column sensitive. Any number of blanks may be inserted before any field.

```
* [comments]
```

Comment records may be placed anywhere in the file.

```
DBCS  number  [comments]
```

This record is optional, but may only be used as the first noncomment record in the file. It identifies the language as being represented in a double-byte character set, and specifies the character set number. The number must be between 0 and 4095. Its value (converted to binary) is used as the high-order half of all non-APL characters in APL Extended Character representation.

```
lang  classno  [comments]
```

This record is used to begin a set of message or keyword definitions. Class 1 is used for messages. Other class numbers are used for keywords, beginning with session manager keywords in class 2. The language class records need not appear in class number order, but after a class is started, all definitions within that class must be given before the next language class record.

Note that “lang” must match the file name, and that “classno” must be a class number that is also available within the base English message/keyword definition. Those class numbers are defined in the AP2XKWD macro.

```
itemno  [fieldno]  ['string']  [comments]
```

This record is used for the first or only line defining an individual message or keyword. The records need not appear in item number order, but an item number may be defined only once within a class group. Messages (but not keywords) may

be built of a number of fields. However, no more than the two fields shown may be defined by one line in the file.

The “itemno” is either a message number (as defined in the AP2XMSGS macro) or a keyword number (as defined in the AP2XKWD macro).

“fieldno” is always omitted for keywords and may be omitted for messages. For messages that begin with a string that is dynamically substituted, it is the message substitution field number. Those numbers are identified in the AP2XMSGS macro.

‘string’ is required when “fieldno” is omitted. It must always be enclosed in apostrophes, as shown. Any imbedded apostrophes must be doubled; for example, ‘DON’’T DO THAT’. When DBCS characters are being used, the enclosing apostrophes must be in EBCDIC. Any DBCS characters used must be bracketed by shift-out (X'0E') and shift-in (X'0F').

Note: APL2 never adds blanks before or after substitution fields or quoted string fields. If two fields in a message should be separated by a blank, the blank must be coded within a quoted string.

```
- [fieldno]   ['string']   [comments]
```

This is used as a continuation for an item record. The hyphen must be coded as shown, and followed by at least one blank. Any substitution field or string field defined here will immediately follow the fields defined in the previous record.

Extending or Adding National Language Help Files

A set of national language help files is distributed with APL2. These files may be tailored by an installation using standard editors. Each file is identified by a 1- to 8-character language name (called "lang" here) that can be assigned to the $\square NLT$ system variable.

Each file is named "lang APL2HELP", and may contain either fixed or variable length records. The file may be on any accessed minidisk; standard search order is used.

At most 72 bytes of each record are used, which means that records may be numbered in columns 73-80, and any such numbers are ignored by APL2.

```
.* [comments]
```

Comment records may be placed anywhere in the file.

```
:HELP.Key
```

Key is any character vector up to and including byte 72 of the record. Trailing blanks in the key are ignored. Blanks occurring before the last nonblank are significant.

A record starting with a :HELP. tag delimits the start of a Help text entry. The entry continues until the next record starting with :HELP or the end of the file.

```
:HELP.Filename.Key
```

Key is any character vector up to and including byte 72 of the record. Trailing blanks in the key are ignored. Blanks occurring before the last nonblank are significant.

Filename is a 1- to 8-character name identifying a name of another file with the same file type. This file contains the Help text entry associated with the Key. The entire contents of the file is used as the entry except for comment records which are ignored.

Consult the comments in the DEFAULT APL2HELP file distributed with APL2 for further details on the format of Help files.

Providing a Local Greeting Message

When users invoke APL2 they first see a set of messages like this sample:

```
    APL2 2.2.00 (English)
    Version 2 Release 2

CLEAR WS
```

The first lines is determined by the system, as is the last line. But the line shown here as “Version 2, Release 2” is taken from a file that you can change at any time. You may provide any number of lines of replacement messages. You might want your users to see this:

```
    APL2 2.2.00 (English)
Call John Smith at 1234 for APL questions or problems.
Maintenance is scheduled for this weekend.

CLEAR WS
```

You can change the messages by editing the file GREETING APL2LANG. Each record in this file is displayed as it appears during APL2 invocation, except that:

- Lines containing an * in column 1 are ignored.
- Trailing blanks are deleted.

Chapter 6. Administering the APL2 Environment

After APL2 has been installed and customized, it requires ongoing attention to operate efficiently and to support APL2 application and end users. This chapter provides information about:

- Running the global SVP service machine
- Running the APL2 port server
- Documenting problems
- Maintaining APL2

Running the Global SVP Service Machine

If you have decided to support the global SVP on your system, you need to know how to use the commands provided to control its operation.

- To start up and initialize the global SVP, log on to the service machine and invoke AP2CSVP with the name of the saved segment, using the command:

  AP2CSVP START *segname maxproc*

  where:

  segname The name of the saved segment defined for the global SVP.

  maxproc The maximum number of users who can be identified to the global SVP at any one time. A minimum of 5 processors is required.

  If installed, the global SVP attempts to identify the user when APL2 is invoked. If it is successful the user remains identified for the duration of the APL2 command, whether or not any global shared variables are used. If it fails because the maximum number of users are already identified, the user is not able to access global shared variables during that APL2 session, even if the number of identified users later drops below the minimum.

  We recommend that the startup of the SVP be included as part of the autolog procedure at system IPL time so that the user ID is automatically logged on whenever the machine is re-IPLed.

- To drop a user from the global SVP, issue the command:

  AP2CSVP SIGNOFF *userid*

  where:

  userid The VM user ID of the user to be dropped.

© Copyright IBM Corp. 1984, 1994

- To determine the current users of the global SVP, issue the command:

  AP2CSVP QUERY *userid*

  where:

  userid An optionally specified VM user ID.

- To terminate the global SVP, issue the command:

  AP2CSVP STOP

Running the Port Server

The port server is an external APL2 function that should be run in a disconnected machine. If you want all users to be able to use cross system shared variables (variables shared across TCP/IP networks), you should install and start the port server.

If you have decided to install the port server, you need to know the commands provided to start and control its operation. The file AP2MPSRV AP2MSAMP is a sample exec for running the port server. A printout of AP2MPSRV is provided in Appendix B, “Sample Files” on page 83.

The port server is called *SERVER*. Although it is typically started using the exec AP2MPSRV, it can be accessed and started from an interactive APL2 session. It is accessed and started as shown below.

```
      3 11 ⎕NA 'SERVER'
1
      SERVER
Enter server port number (default 31415):
Enter server password: SECRET
```

Figure 6. Running the port server

The server prompts for the port number it should use. If no response is given, it defaults to using 31415. If a port other than 31415 is given, then users on the same system should use the APNAMES invocation option to inform AP 119 of the port number. For further details, see *APL2/370 Programming: System Services Reference*.

If a port other than 31415 is given, then users on remote systems need to specify the port number in their TCP/IP profile files or use the AP 119 SETLPORT command. For further details see *APL2/370 Programming: System Services Reference*.

The server also prompts for a password that is required of users attempting to use restricted server commands. If no response is given, no restricted server commands can be used. The following restricted server commands are provided for administration of the port server. Consult the documentation of AP 119 in *APL2/370 Programming: System Services Reference* for information about their use.

PSLIST List the port server's table of registered users.

PSCLEAR Clear the port server's table of registered users.

PSSHUTD Shut down the port server.

Note: If the port server is run in a disconnected machine using the sample exec, the output of the PSLIST command is sent to the APL2 session log if it exists. No result is visible until the user ID is reconnected and the port server is shut down.

The APL2 invocation option RUN can also be used to start the port server. In this case, the INPUT option would typically be used to supply the prompt responses. Examine the AP2MPSRV exec for a sample of using the RUN and INPUT parameters.

The port server maintains a table of active users of cross systems shared variables. This table is stored in the workspace of the APL2 session under which the port server is running. If a very large number of users are active, it is possible for the port server to hit a workspace full condition. If the port server hits a workspace full condition, it issues a *WS FULL* to it's session log. However, it does not terminate.

If users are unable to share cross systems shared variables, and there are very many active users, shutdown the port server and examine the session log. If one or more *WS FULL* messages were produced, use the WSSIZE invocation option in AP2MPSRV to increase the server's workspace size and restart the server.

If the server is run with a port number other than the default 31415, and if the default APNAMES invocation option was not changed to specify the port number, users should be informed of the port number.

To start the port server using AP2MPSRV, perform the following steps:

1. Edit the AP2MPSRV exec appropriately. (Rename it to AP2MPSRV EXEC.) Follow the instructions in the file. Be sure to provide a password or else you will not be able to shut down the port server except by IPLing the machine. A printout of AP2MPSRV is provided in “AP2MPSRV EXEC Sample — AP2MPSRV AP2MSAMP” on page 91.

   If a port number other than the default, 31415, is used, be sure to notify the users of the fact.

   If the name of the APL2 module was changed, also modify this file so that it uses the correct name.

2. Verify the functioning of the port server:

   a. Start the port server by executing the AP2MPSRV exec from the APL2PSRV user ID.

   b. Initiate an APL2 session on another user ID, that by issuing a port server command, verifies the port server is operating correctly. Invoke APL2 using the invocation option DEBUG(1) to cause secondary messages to be automatically displayed. If a port number other than the default, 31415, was used in the AP2MPSRV exec, also use the invocation option APNAMES(AP2X119 SERVPORT(*portnumb*)) when invoking APL2. *portnumb* must be the port number you specified in AP2MPSRV.

      During this process, if APL2 is not able to contact the port server, auxiliary processor 119 should issue one or more messages describing the problem.

If APL2 does not return the correct response or an error occurs, exit APL2 by entering:)*OFF*

- Once in APL2, enter:

```
119 ⎕SVO 'C119'
```

  The system should return a `1`.

- Enter:

```
1 ⎕SVC 'C119'
```

  The system should return `1 1 1 1`.

- Again enter:

```
119 ⎕SVO 'C119'
```

  The system should return a `2`. If it does not, hit enter a few times and try again.

- Enter:

```
C119←'AP' 'PSSHUTD' 'password'
C119
```

  Where:

  password is the port server password you specified in AP2MPSRV.

  The system should return `0 0 0`

- Exit APL2 by entering:

```
)OFF
```

c. Determine if the server has shut down (the APL2PSRV user ID is in APL2 immediate execution mode.)

   If it has not shut down, shut down the server manually by logging on to the APL2PSRV user ID, reconnecting by typing BEGIN, signalling interrupt, and entering:

```
)RESET
)OFF
```

d. Exit APL2 on the other user ID by entering:

```
)OFF
```

If the port server correctly shut down in response to the *PSSHUTD* command, you have verified it operates correctly.

If the port server did not correctly shut down in response to the *PSSHUTD* command, you should try running the port server from an interactive APL2 session rather than in a disconnected machine. In an interactive session you are to see if the port server issues any error messages. Starting the port server interactively is discussed in at the beginning of this section.

3. Start the port server again by executing the AP2MPSRV exec.

Documenting Problems

Detailed procedures for diagnosing and reporting problems are found in *APL2/370 Diagnosis Guide*. You can shorten the time required to resolve a problem by gathering relevant documentation before you report the incident. The following serve to highlight some of the general techniques useful in diagnosing APL2 problems.

- What was the environment?

  Whenever you report a problem, you should be able to describe the level of VM, APL2, and related products (such as GDDM) that were in use at the time of the failure. You might want to know what parts of APL2 were in use at the time of the failure. Chapter 1, "Overview of APL2 and the Installation Process" on page 1 contains a general description of the basic components of an APL2 system.

- Is it reproducible?

  Perhaps the single most valuable piece of information in diagnosing problems is the minimum set of conditions to reproduce the failure. A problem that you can recreate can usually be solved quickly.

- Do I have a record?

  When a problem occurs intermittently, it can often be diagnosed by comparing the CP console files and/or APL session manager logs with previous occurrences. You should retain the records of your session along with any dumps or damaged workspaces.

- Should I get a dump or trace?

  Dumps and traces are sometimes the only way to diagnose intermittent problems. APL2 provides several forms of dumps and traces you can use. For more information, see *APL2/370 Diagnosis Guide*.

Maintaining APL2

The following information is intended to provide an overview of the maintenance process. Whenever you apply preventive service, read and follow the instructions provided with the Program Temporary Fix (PTF) tape. Whenever you apply corrective service, follow the instructions given below.

Application of maintenance to APL2 is merely another instance of customizing APL2. Customizing APL2 relies heavily on the NAMEFIND facility, as described in *VM CMS Command and Macro Reference*. For a description of how APL2 uses NAMEFIND and NAMES files through APL2's AP2M EXECs to customize APL2, see "Customizing Your APL2 Installation Process" on page 46.

Preventive Service with PUT

Preventive service, service distributed to all customers on a regular basis, comes in a format acceptable to the VM PUT process and is applied by the 5688228B EXEC (for APL2), or by the 5688229B EXEC (for APL2 Application Environment), which is distributed on the VM PUT tape. In VM, preventive service is cumulative; you need only the original distribution tape and the latest preventive service tape to construct a system at the most up-to-date level.

Corrective Service

Corrective service consists of PTFs distributed to a specific customer to correct a specific problem. PTFs come on an unlabeled tape without an automatic installation procedure. In general, the instructions on the tape ask you to execute one or more of the maintenance EXECs in order to apply the corrective service.

After all the files in this PTF have been read from the tape, you must regenerate APL2 for CMS. If you want to resave any discontiguous shared segments in the process, your virtual machine must be larger than the largest saved segment end address and you must have privilege class C or E. In any case you should have at least 4 megabytes of virtual storage.

To regenerate APL2, you should:

1. LINK and ACCESS the disk where APL2 is regenerated.

   By default APL2 is regenerated on your A-disk. You can regenerate APL2 on a different disk by assigning a different filemode letter to the EXEC2 variable &-FM in the :CMD. tag of the PROLOG-AP2MAINT :NICK. in an AP2LOCAL NAMES file. See “Customizing Your APL2 Installation Process” on page 46 for additional information on how the APL2 maintenance EXECs reference NAMES files during their execution, and how you can use NAMES files to control the customization of APL2.

2. If the files for this PTF were read to a different disk than where APL2 is regenerated, copy them to the disk where APL2 is regenerated.

3. If the disk where APL2 is regenerated is different from the disk where the APL2 product is installed, LINK and ACCESS the disk where APL2 is installed.

4. If any files with the file type AP2MSAMP are supplied in this PTF, compare them to the originals supplied at installation time and to the actual versions you may be using to see what changes have been made. If appropriate, make similar changes in your versions. See “Changing the APL2 System and Invocation Options” on page 62.

5. If any files with the filetype TEXT are supplied in this PTF, the APL2 LOADLIBs and MODULEs must be regenerated.

   a. To regenerate the utility MODULEs, enter:

      ```
      AP2MAINT UTIL-PTF
      ```

   b. To regenerate the main MODULEs and LOADLIBs, enter:

      ```
      AP2MAINT APL2-PTF
      ```

      or, for the Application Environment:

      ```
      AP2MAINT APL2AE-PTF
      ```

c. If the LOADLIBs are to be resaved in saved segments, enter:

```
AP2MAINT APL2SEG
```

or, for the Application Environment:

```
AP2MAINT APL2AESEG
```

d. If the global SVP is installed, also enter:

```
AP2MAINT CSVP-PTF
```

Note: List items 5c and 5d require the following consideration:

Before being saved, segments must first be defined. If already defined, they might need to be PURGEd and redefined. See "Customizing Saved Segment Information for APL2" on page 52 for information on automating this process.

6. If any files with the file type COPY are supplied in this PTF, the APL2 maclibs must be regenerated:

   ```
   AP2MAINT APL2-MAC
   ```

7. If a file AP2F3800 CNTRL is supplied in this PTF and you have installed the APL2 3800 line-mode fonts, the fonts must be regenerated. To do this, enter:

   ```
   AP2MAINT GEN3800
   ```

   Privileged class A, B, or C is required to regenerate the fonts.

8. If any files with the file type ASMSQL are supplied in this PTF, and you use APL2 to access SQL/DS, your SQL/DS database administrator must pre-process the files using the SQL/DS EXEC SQLPREP. The corresponding text decks supplied, if any, have already been preprocessed. The output from the SQLPREP EXEC does not need to be assembled.

9. If APL2 has been regenerated with this PTF onto a disk different from the production disks, use the AP2MCOPY EXEC to copy files to the production disks. For more information, see "Step 6–Copy Files to Production Disks (Optional)" on page 21.

10. Run the verification procedure. For more information on how this is done, refer to "Step 6–Copy Files to Production Disks (Optional)" on page 21.

11. Update local greeting message if desired.

Appendix A. Sample User IDs and Disk Tables

Sample Installation User ID

```
USER APL2 INSTPSWD  4M   8M   BEG
 ACCOUNT 9999 I5668899
 IPL CMS
 CONSOLE 009 3215 T
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A
 LINK CMSMAINT 190 190 RR
 LINK CMSMAINT 19E 19E RR
 MDISK 191 3350    210  060  SYS003 MR ALL
```

Any user ID may be used to install APL2, but IBM recommends that each release of APL2 be installed on a unique user ID, or at least on a unique mini-disk.

By default APL2 is installed on the installer's A-disk.

To save segments the installer must have appropriate CP privilege and have a large enough virtual machine.

To install 3800 line-mode fonts the installer must have appropriate CP privilege.

Sample APL2 User User ID

```
USER APLUSER  USERPSWD  4M   4M   G
 ACCOUNT 9999 APL2
 IPL CMS
 CONSOLE 009 3215 T
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A
 LINK CMSMAINT 190 190 RR
 LINK CMSMAINT 19E 19E RR
 MDISK 191 3330    105  005  USER06 MR
```

Any general user may use APL2.

Sample Global SVP User ID

```
USER AP2SVP  GSVPPSWD 2M   8M   EG
 ACCOUNT 9999 APL2-SVP
 IPL CMS
 CONSOLE 009 3215 T
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A
 LINK CMSMAINT 190 190 RR
 LINK CMSMAINT 19E 19E RR
 MDISK 191 3330    176  001  SYS004 MR
```

Some installations prefer to give the GSVP user ID CP privilege to save the GSVP saved segment, but that is optional.

© Copyright IBM Corp. 1984, 1994

Include the following two commands in the global SVP Profile Exec:

```
SET AUTOREAD OFF
CP SET RUN ON
```

Sample Port Server User ID

```
USER APL2PSRV  PSRVPSWD  4M   4M   G
 ACCOUNT 9999 APL2
 IPL CMS
 CONSOLE 009 3215 T
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A
 LINK CMSMAINT 190 190 RR
 LINK CMSMAINT 19E 19E RR
 MDISK 191 3330    105  005  SYS243 MR
```

The port server user ID needs no special CP privilege.

Disk Tables for Selected CMS Devices

The A-disk should have the following space:

Figure 7. A-Disk Space Requirements for APL2

APL2 Material	APL2 Space Requirements	Application Environment Space Requirements
Basic materials	13,600 1K or 3,600 4K blocks	7,200 1K or 1,900 4K blocks
Optional text and macros	5,500 1K or 1,700 4K blocks	4,000 1K or 1,300 4K blocks
Optional 3800 fonts	4,200 1K or 1,200 4K blocks	4,200 1K or 1,200 4K blocks

In planning for disk space, larger block sizes are often preferred when disks contain large files like LOADLIBs and workspaces, because fewer disk operations are needed to read or write them. Smaller block sizes are used when disks have many small files with sizes less than a block.

Figure 8. Blocks/Cylinder—CMS Disk Format

Device	1K Block	4K Block
3330	209	57
3340	84	24
3350	450	120
3375	300	96
3380	465	150
3390	558	180

Appendix B. Sample Files

AP2EXIT EXEC Sample — AP2EXIT AP2MSAMP

```
|           &TRACE OFF
|           &VERB = &1
|           &UID = &2
|           &LIB = &3
|           &TERM = &4
|           &DEBUG = &5
|           &FUNC = &SUBSTR OF &VERB 1 3
|           &IF .&DEBUG = . &DEBUG = 0
|           &JUNK = &DATATYPE OF &DEBUG
|           &IF .&JUNK  NE .NUM &DEBUG = 0
|           &IF .&FUNC = .ERR &GOTO -OFFAPL
|           &IF .&FUNC = .OFF &GOTO -OFFAPL
|           &IF .&FUNC = .ON  &GOTO -ONAPL
|           &BEGTYPE -ETELL
| *
| * AP2EXIT  - APL2 Invocation/Termination EXEC
| *            Responsible for Changing/Restoring Environment
| *
| * FORMAT is VERB USERID LIBNO TERMTYPE
| *
| *  VERB     - ON, OFF, ERR
| *  USERID   - CMS userid
| *  LIBNO    - APL2 Signon Library  (Default 1001)
| *  TERM     - GRAF, LINE, NONE, BATCH
| *             NOTE: If TERM is NONE, this EXEC should not
| *                   affect the screen, since NONE indicates that
| *                   either the machine is running disconnected
| *                   or APL2 has been invoked from an environment
| *                   that controls the screen, not to be disturbed.
| *  DEBUG    - Value of Invocation Debug Parameter
| *
| * Example:  ON MANGLIN 00001001 GRAF 00000000
| *
| ************************************************************
| *  APL2 Version 2 Release 2
| *  Licensed Materials - Property of IBM
| *  5688-228, 5688-229 (C) COPYRIGHT IBM CORP. 1984, 1994.
| *  See Copyright Instructions
| ************************************************************
|           -ETELL
|           &EXIT 100
|           *
|           *  APL2 IS INITIALIZING
|           *
| -ONAPL
|           &MSGSW = &DIVISION OF &DEBUG 2
|           &MSGSW = &DEBUG - &MSGSW - &MSGSW
```

© Copyright IBM Corp. 1984, 1994

```
|            &MSG = &STRING OF 0 &ARGSTRING
|            &MSGNUM = &CONCAT OF 0 665
|            &MSGNUM = &LEFT OF &MSGNUM 12
|            &IF &MSGSW ¬= 0 &IF &TERM ¬= NONE &TYPE &MSGNUM &MSG
|            * Remove FILEDEF for APLDUMP if dumps not wanted
|            FILEDEF APLDUMP PRINTER
|            * Add FILEDEF for AP2LOAD if APNAMES parameter required
|           *FILEDEF AP2LOAD DISK APL2 LOADLIB *
|            * Add GLOBAL TXTLIB here if GDDM not in Saved Segment
|           *GLOBAL TXTLIB ADMPLIB ADMGLIB
|            &IF &TERM = LINE CP SET LINEDIT OFF
|            &IF &TERM NE BATCH &IF &TERM NE NONE CP TERMINAL APL ON
|            &IF &TERM = LINE CP TERMINAL ATTN OFF
|            *
|            * (Note: EXIT at initialization with RC 1 causes session to be
|            * terminated immediately.)
|            *
|            &EXIT 0
|            *
|            *  APL2 IS TERMINATING
|            *
|  -OFFAPL
|            * Remove FILEDEFs as appropriate
|            FILEDEF APLDUMP CLEAR
|           *FILEDEF AP2LOAD CLEAR
|            * Add GLOBAL TXTLIB here to free GDDM libraries if not in DCSS
|           *GLOBAL TXTLIB
|            &IF &TERM = LINE CP SET LINEDIT ON
|            &IF &TERM NE BATCH &IF &TERM NE NONE CP TERMINAL APL OFF
|            &IF &TERM = LINE CP TERMINAL ATTN ON
|            *
|            * RC at exit is ignored
|            *
|            &EXIT 0
```

AP2LOCAL NAMES Sample — AP2LOCAL AP2MSAMP

```
| :rem.*-------------------------------------------------------------*
| *
| *  Sample AP2LOCAL NAMES file
| *
| *-------------------------------------------------------------*
| *
| *  APL2 Version 2 Release 2
| *  Licensed Materials - Property of IBM
| *  5688-228, 5688-229 (C) COPYRIGHT IBM CORP. 1984, 1994.
| *  See Copyright Instructions
| *
| *-------------------------------------------------------------*
| *
| *  The installation/maintenance process is to follow the tree
| *  specified by some list of :NICK tags.
| *
| *  Although the AP2Mnnn NAMES file contains a complete description
| *  of the product (where "nnn" is the system level begin used by
| *  the AP2MAINT EXEC), AP2LOCAL NAMES file is searched first
| *  for each :NICK encountered by AP2MAINT or AP2MLNK EXECs
| *  and thus can be used to customize the product without
| *  altering the file supplied by IBM.
| *
| *  If you choose to front-end a :NICK, remember you are
| *  replacing the ENTIRE entry, not just individual tags.
| *
| *-------------------------------------------------------------*

| :rem.*-------------------------------------------------------------*
| *
| *  Sample showing filemodes for loading APL2 PROGRAM PRODUCT 5688-228
| *  install tape and service tapes.
| *
| *  Each letter or '_' specifies the filemode for a tape file.
| *
| *  On the install tape, there are nine tape files:
| *
| *     1 - Install EXEC
| *     2 - Memo to Users
| *     3 - Install materials and samples
| *     4 - HELP files
| *     5 - Run-time files
| *     6 - Workspaces
| *     7 - Optional TEXT, MACROs, Workspace Images
| *     8 - Optional Migration Aids for VS APL APs
| *     9 - Optional 3800 Fonts
| *
| *  If '_' is specified, the file is skipped.
| *
| *  By the time the :Nick.I5688228 is examined by the I5688228 EXEC
| *  the first two tape files will have already been loaded.  The
```

```
| *  EXECs will not reload the first two tape files.  The
| *  letters for the first two tape files serve only as place holders,
| *  so that the remaining letters are aligned with the corresponding
| *  tape files.
| *
| *  The I5688228 nick shown below loads the third through sixth files
| *  from tape to the A-disk.  The optional files are not loaded.
| *
| *  :Nick.5688228B works in a similar manner with the 5688228B service
| *  EXEC.  Service tapes have only three tape files.  All the service
| *  is combined in the third tape file.  Initially all service is
| *  loaded to the filemode specified for the third tape file.  If
| *  other tapefiles on the install tape were loaded to different
| *  filemodes, you can specify those filemodes here and the service
| *  exec will move the service for those tapefiles to the specified
| *  filemodes.
| *
| *  The 5688228B nick shown below loads the third file
| *  from tape to the A-disk.
| *
| *-------------------------------------------------------------------*

| :Nick.I5688228         :VMFPLC2-FMs.xxAAAA___

| :Nick.5688228B         :VMFPLC2-FMs.xxA______

| :rem.*------------------------------------------------------------*
| *
| *  Sample showing filemodes for loading the APL2 EXECUTE-ONLY SUBSET
| *  PROGRAM PRODUCT 5688-229 install tape and service tapes.
| *
| *  These tags work in the same manner as those of 5688-228, except
| *  that there are seven tape files:
| *
| *     1 - Install EXEC
| *     2 - Memo to Users
| *     3 - Install materials and samples
| *     4 - HELP files
| *     5 - Run-time files
| *     6 - Optional TEXT, MACROs, Workspace Images
| *     7 - Optional 3800 Fonts
| *
| *-------------------------------------------------------------------*

| :Nick.I5688229         :VMFPLC2-FMs.xxAAA__

| :Nick.5688229B         :VMFPLC2-FMs.xxA____

| :rem.*------------------------------------------------------------*
| *
| *  Each :NICK will be searched for the first tag (in order) of:
| *
| *            :cmd    - A Command to be executed immediately
| *                      The ';' may be used to delimit clauses.
```

```
| *                     If a non-zero return code occurs,
| *                     the active :NICK is abandoned.
| *                     NOTE: This tag may be any EXEC2
| *                           statement executable with &READ 1
| *
| *           :msg    - Arbitrary message to type
| *                     The ';' may be used to delimit lines.
| *
| *           :part   - Name of a TEXT deck for a module
| *                     The ';' may be used to delimit lines.
| *
| *           :list   - List of additional :NICKS to process
| *
| *--------------------------------------------------------------------*

| :rem.*---------------------------------------------------------------*
| *
| *  Sample showing AP2MAINT PROLOG and EPILOG
| *
| *  They could be used do things like ACCESS/FREE disks
| *
| *  The ':cmd' below sets a global variable to the filemode
| *  to be used for commands issued by AP2MAINT.
| *
| *  The ':list' below suppresses 3800 font installation,
| *  but includes GDDM and SQL support.  The LOCAL in the list tells
| *  the EXECs that there may be local text decks to be added.
| *
| *---------------------------------------------------------------------*

| :Nick.PROLOG-AP2MAINT  :cmd.&-FM = A2
|                        :rem.*** AP2MAINT Started - &ARGSTRING
|                        :list. GDDM SQL NOF3800 LOCAL
| :Nick.EPILOG-AP2MAINT  :rem.*** AP2MAINT Ended

| :rem.*---------------------------------------------------------------*
| *
| *  Sample showing how to override IBM variables
| *
| *  The IBM tag for SEGS-cmd is SAVESEGS which causes the segments
| *  to actually be saved.
| *  The alternative is TESTSEGS which leaves segments as NUCEXT only
| *  so new level of system can be tested.
| *
| *  (If you have not defined segments, this tag is irrelevant.)
| *
| *---------------------------------------------------------------------*

| :Nick.SEGS-cmd         :cmd.&-SEGS = SAVESEGS

| :rem.*---------------------------------------------------------------*
| *
| *  Sample showing saved segment declarations for APL2
| *
```

```
| *  The IBM default is to save only d-SEG1 and d-HIGH1
| *  If you want to have one AP2HIGH above the line and another
| *  below, you would add d-HIGH2 to the list as shown.
| *
| *--------------------------------------------------------------------*

| :Nick.APL2SEG      :rem. APL2 Saved Segments
|                    :list. clean-SEG1 d-SEG1 clean-HIGH1 d-HIGH1
|                           clean-HIGH2 d-HIGH2

| :Nick.APL2AESEG    :rem. APL2AE Saved Segments
|                    :list. clean-AESEG1 d-AESEG1 clean-AEHIGH1 d-AEHIGH1
|                           clean-AEHIGH2 d-AEHIGH2

| :rem.*-------------------------------------------------------------*
| *
| *  You must also change the names of any segments you have changed
| *  in your local modification to AP2VIOPT (if any).
| *
| *--------------------------------------------------------------------*

| :Nick.d-SEG1     :cmd.EXEC AP2MLNK AP2V22V1 &-SEGS   APL2 APL2 (&-Sys
| :Nick.d-AESEG1   :cmd.EXEC AP2MLNK AP2V22E1 &-SEGS   APL2 APL2AE (&-Sys
| :Nick.d-HIGH1    :cmd.EXEC AP2MLNK AP2V22W1 &-SEGS AP2HIGH AP2HIGH
|                       (&-SYS
| :Nick.d-AEHIGH1  :cmd.EXEC AP2MLNK AP2V22F1 &-SEGS AP2HIGH AP2HIAE
|                       (&-SYS
| :Nick.d-HIGH2    :cmd.EXEC AP2MLNK AP2V22W2 &-SEGS AP2HIGH AP2HIGH
|                       (&-SYS
| :Nick.d-AEHIGH2  :cmd.EXEC AP2MLNK AP2V22F2 &-SEGS AP2HIGH AP2HIAE
|                       (&-SYS

| :rem.*-------------------------------------------------------------*
| *
| *  The "Clean-" tags provide a place to automate the PURGE and DEFSEG
| *  commands associated with the shared segments. As shipped in AP2M220
| *  NAMES, these tags are empty.  You can add your PURGE and/or
| *  DEFSEG commands so that each time you call the APL2SEG or APL2AESEG
| *  tags they will automatically be issued.  Note that the user id
| *  issuing the APL2SEG or APL2AESEG must have sufficient authority to
| *  issue any commands placed in the "Clean-" tags.
| *
| *  To use these samples, change the ":rem." to ":cmd.", change the
| *  segment addresses to match those on your system, and if the segment
| *  names have been changed above, change them here as well.
| *
| *--------------------------------------------------------------------*

| :Nick.Clean-SEG1    :rem.CP PURGE NSS NAME AP2V22V1 ;
|                          CP DEFSEG AP2V22V1 600-64F SR
| :Nick.Clean-HIGH1   :rem.CP PURGE NSS NAME AP2V22W1 ;
|                          CP DEFSEG AP2V22W1 6000-60FF SR
| :Nick.Clean-HIGH2   :rem.CP PURGE NSS NAME AP2V22W2 ;
```

```
|                          CP DEFSEG AP2V22W2 500-5FF SR
|  :Nick.Clean-AESEG1  :rem.CP PURGE NSS NAME AP2V22E1 ;
|                          CP DEFSEG AP2V22E1 600-64F SR
|  :Nick.Clean-AEHIGH1 :rem.CP PURGE NSS NAME AP2V22F1 ;
|                          CP DEFSEG AP2V22F1 6000-60FF SR
|  :Nick.Clean-AEHIGH2 :rem.CP PURGE NSS NAME AP2V22F2 ;
|                          CP DEFSEG AP2V22F2 500-5FF SR
|  :Nick.Clean-SMSEG   :rem.CP PURGE NSS NAME AP2SM22 ;
|                          CP DEFSEG AP2SM22 7000-7FFF SW

|  :rem.*-----------------------------------------------------------------*
|  *
|  *  Sample showing how to tell the APL2 CMSIVP package about saved
|  *  segments belonging to GDDM and SQL/DS.
|  *
|  *  The GDDMSEGS and SQLSEGS :NICKS are used to tell CMSIVP about the
|  *  saved segments used on your system for the GDDM and SQL/DS products.
|  *  If a :LIST tag is left empty, the CMSIVP will not test those parts
|  *  of APL2 which make use of the other product.
|  *
|  *  If you want the CMSIVP to test the parts of APL2 which do use GDDM
|  *  or SQL/DS, you must identify in the :LIST tag the names of the saved
|  *  segments which the product uses.  For example, SQL/DS normally uses
|  *  the segment name SQLRMGR for the shared segment used by APL2 to
|  *  access SQL/DS.  If your installation has installed SQL/DS with
|  *  the default segment names, and you wish to verify APL2 access to
|  *  SQL/DS, change:
|  *
|  *      :Nick.SQLSEGS      :list.
|  *
|  *  to
|  *
|  *      :Nick.SQLSEGS      :list.SQLRMGR
|  *
|  *  If you want the CMSIVP to test the parts of APL2 which do use GDDM
|  *  or SQL/DS, and GDDM or SQL/DS is not installed in a saved segment
|  *  on your system, specify NONE as the value for the :list tag.  For
|  *  example
|  *
|  *      :Nick.SQLSEGS      :list.NONE
|  *
|  *-----------------------------------------------------------------*

|  :Nick.GDDMSEGS     :list.
|  :Nick.SQLSEGS      :list.

|  :rem.*-----------------------------------------------------------------*
|  *
|  *  Sample showing how to add local text decks to the APL2 module.
|  *  The sample checks for the APL2 options module, and adds it if
|  *  it exists.
|  *
|  *-----------------------------------------------------------------*
```

```
| :Nick.Local-TEXT     :rem. Make local Options modules active           .
|                      :list.Local-OPT
|                            Local-SCOPT Local-APs Local-EPs

| :Nick.Local-HIGHTEXT :rem. Make local Options modules active           .
|                      :list.Local-OPT local-V127I
|                            Local-HIAPs Local-HIEPs

| :Nick.Local-AETEXT   :rem. Make local Options modules active           .
|                      :list.Local-VIOPE
|                            Local-APs Local-EPs

| :Nick.Local-HIAETEXT :rem. Make local Options modules active           .
|                      :list.Local-VIOPE local-V127I
|                            Local-HIAPs Local-HIEPs

| :Nick.Local-OPT    :cmd.STATE AP2VIOPT TEXT *
|                    :Part.AP2VIOPT TEXT
|                    :msg.+++ Local APL2 options (AP2VIOPT) included

| :Nick.Local-SCOPT  :rem.* You could duplicate the Local-OPT nick here,
|                    and change it to check for APLSCOPT.

| :Nick.Local-APs    :rem.* Add your APs here in ':Part.' and/or ':List'

| :Nick.Local-HIAPs  :rem.* Add your APs here in ':Part.' and/or ':List'
|                    :rem.* APs in the HIGH loadlib must not have any
|                    :rem.* dependencies requiring them to run below
|                    :rem.* the 16M line. They will be started in
|                    :rem.* 31-bit mode if on an XA or ESA system.

| :Nick.Local-EPs    :rem.* Add your EPs here in ':Part.' and/or ':List'

| :Nick.Local-HIEPs  :rem.* Add your EPs here in ':Part.' and/or ':List'
|                    :rem.* EPs in the HIGH loadlib must not have any
|                    :rem.* dependencies requiring them to run below
|                    :rem.* the 16M line. They will be started in
|                    :rem.* 31-bit mode if on an XA or ESA system.
```

AP2MPSRV EXEC Sample — AP2MPSRV AP2MSAMP

```
/*REXX ***************************************************************/
/*                                                                    */
/*  Exec to run the APL2 Port Server on CMS.                          */
/*                                                                    */
/*********************************************************************/
/*  APL2 Version 2 Release 2                                          */
/*  Licensed Materials - Property of IBM                              */
/*  5688-228, 5688-229 (C) COPYRIGHT IBM CORP. 1992, 1994.            */
/*  See Copyright Instructions                                        */
/*********************************************************************/
Trace o
port_number = "31415"     /* Number of listening port used by server */
password    = "SECRET"    /* Authorized commands password            */

input = "'"port_number"' '"password"'"
"CP SPOOL CONSOLE TO * START"
"CP SET RUN ON"
"CP DISC"
"APL2 DEBUG(35) RUN(SERVER) INPUT("input")"
return rc
```

Sample Exit for Global SVP — AP2SVPEX AP2MSAMP

This appendix contains Product-sensitive Programming Interface and Associated Guidance Information.

```
SVPEX    TITLE 'Sample APL2/CMS GSVP Installation Exit'                 00010000
********************************************************************** 00020000
*                                                                    * 00030000
* MODULE NAME = AP2SVPEX                                             * 00040000
*                                                                    * 00050000
* DESCRIPTIVE NAME = SAMPLE APL2/CMS GSVP Installation EXIT          * 00060000
*                                                                    * 00070000
* COPYRIGHT:  Licensed Materials - Property of IBM                   * 00072990
*                                                                    * 00074980
*             5688-228, 5688-229                                     * 00077960
*             (C) COPYRIGHT IBM CORP. 1984, 1994                     * 00079940
*                                                                    * 00080950
*             See Copyright Instructions                             * 00082940
*                                                                    * 00084930
* STATUS:     APL2 VERSION 2 RELEASE 2                               * 00085930
*                                                                    * 00086930
* FUNCTION = This code resides as a CMS NUCLEUS EXTENSION in order   * 00090000
*            to invoke the SVPEXIT EXEC when a user requests         * 00100000
*            the use of the GLOBAL SVP in APL2/CMS.                  * 00110000
*            The EXEC contains any installation-specific processing. * 00120000
*            The Help text (See label TELLTEXT) explains the specific * 00130000
*            requirements that the EXEC must meet.                   * 00140000
*                                                                    * 00150000
*      DEPENDENCIES = VM/SP CMS Release 5                            * 00167970
*                                                                    * 00170000
*      REFERENCES   =                                                * 00180000
*                .SC19-6203 VM/SP System Programmer's Guide          * 00190000
*                   Writing CMS commands, VMCF, Debugging            * 00200000
*                .SC19-6209 VM/SP CMS Command and Macro Reference    * 00210000
*                   Using CMS command, Nucleus Extensions            * 00220000
*                                                                    * 00230000
* MODULE TYPE = BAL                                                  * 00240000
*                                                                    * 00250000
*      PROCESSOR = ASSEMBLER H                                       * 00260000
*                  .The Following CMS commands create the module     * 00270000
*                   GLOBAL MACLIB CMSLIB DMSSP DMKSP                 * 00280000
*                   HASM AP2SVPEX                                    * 00290000
*                   LOAD AP2SVPEX (ORIGIN TRANS                      * 00300000
*                   GENMOD AP2SVPEX                                  * 00310000
*                                                                    * 00320000
*      ATTRIBUTES = CMS TRANSIENT relocated to CMS NUCLEUS storage   * 00330000
*                   and thus must have no address constants.         * 00340000
*                   Disabled, Supervisor state, and Key 0.           * 00350000
*                   Called while processing external interrupt       * 00360000
*                   and thus must NOT use VMCF,IUCV, timers, etc.    * 00370000
*                                                                    * 00380000
* ENTRY POINT = AP2SVPEX (then NUCSVPEX after relocation)            * 00390000
*                                                                    * 00400000
*      LINKAGE = Via SVC 202 from AP2CSVP (APL2 GSVP Control)        * 00410000
```

```
|  *               as a function (x'01' in byte 0 of R1).              * 00420000
|  *                                                                   * 00430000
|  * INPUT = R0  CMS Extended Plist                                    * 00440000
|  *             EPLOPT points at VMCF Message Head (VMCMHDR)          * 00450000
|  *                    VMCMMID = 0 or proposed account number         * 00460000
|  *                    VMCMUSER= VM userid to validate                * 00470000
|  *          R1  CMS Tokenized Plist                                  * 00480000
|  *          R2  SCBLOCK address (for nucleus extension)              * 00490000
|  *          R13 CMS Save Area (12 Doublewords)                       * 00500000
|  *          R14 CMS Return address                                   * 00510000
|  *          R15 Entry Address                                        * 00520000
|  *                                                                   * 00530000
|  * OUTPUT = R15 is Return Code                                       * 00540000
|  *           For invocation by AP2CSVP                               * 00550000
|  *              -1 = SVPEXIT EXEC not found (Authorize if ID ¬= 0)   * 00560000
|  *               0 = User is Authorized with ID set in VMCMMID       * 00570000
|  *               1 = User is not Authorized but ID is set in VMCMMID * 00580000
|  *           Otherwise:                                              * 00590000
|  *               0 = Installed OK                                    * 00600000
|  *              24 = Invalid Operand                                 * 00610000
|  *             100 = HELP text printed                               * 00620000
|  *                                                                   * 00630000
|  *                                                                   * 00640000
|  * MACROS =                                                          * 00650000
|  *           REGEQU   - PROVIDE REGISTER EQUATES                     * 00660000
|  *           SCBLOCK  - NUCLEUS EXTENSION CONTROL                    * 00670000
|  *           VMCBLOKS - VMCF MESSAGES  (FROM DMKSP MACLIB)           * 00680000
|  *                                                                   * 00690000
|  *                                                                   * 00700000
|  ********************************************************************* 00710000
|           SPACE 2                                                     00720000
|           MACRO                                                       00730000
|  &L       CMD   &TEXT                                                 00740000
|  .*                                                                   00750000
|  .*       MACRO TO ISSUE CMS COMMAND AND SET CC BASED ON RC           00760000
|  .*                                                                   00770000
|  &L       LA    R1,&TEXT                                              00780000
|           SVC   CMS202                                                00790000
|           DC    AL4(1)                                                00800000
|           LTR   R15,R15             SUCCESS?                          00810000
|           MEND                                                        00820000
|           SPACE 2                                                     00830000
|           MACRO                                                       00840000
|           TELL  &TEXT                                                 00850000
|  .*                                                                   00860000
|  .*       MACRO TO GENERATE CONSTANTS FOR HELP MESSAGE                00870000
|  .*                                                                   00880000
|           LCLB  &PAIR                                                 00890000
|           LCLA  &I,&LEN                                               00900000
|  &I       SETA  1                                                     00910000
|  &LEN     SETA  K'&SYSLIST(1,1)-2                                     00920000
|  &PAIR    SETB  0                                                     00930000
|  .QLOOP1  ANOP  ,                   LOOP TO LOOK FOR IMBEDDED QUOTES  00940000
|  &I       SETA  &I+1+&PAIR                                            00950000
```

```
|          AIF   (&I GE K'&SYSLIST(1,1)).QDONE1                          00960000
|  &PAIR   SETB  ('&SYSLIST(1,1)'(&I,2) EQ '''''' OR '&SYSLIST(1,1)'(&I,2X00970000
|                ) EQ '&&')                                              00980000
|  &LEN    SETA  &LEN-&PAIR                                              00990000
|          AGO   .QLOOP1                                                 01000000
|  .QDONE1 ANOP                                                          01010000
|          DC    AL1(&LEN),C&TEXT                                        01020000
|          MEND                                                          01030000
|          EJECT 1                                                       01040000
|  ********************************************************************* 01050000
|  *                                                                   * 01060000
|  *    INITIAL ENTRY POINT - HANDLE NUCLEUS EXTENSION HOUSEKEEPING    * 01070000
|  *                                                                   * 01080000
|  ********************************************************************* 01090000
|  AP2SVPEX CSECT                                                        01100000
|          USING WORK,R13                                                01110000
|          STM   R14,R2,WR14         ENTRY REGISTERS                     01120000
|          LR    R12,R15             ADDRESSABILITY                      01130000
|          USING AP2SVPEX,R12                                            01140000
|          B     BYID               - AROUND IDentifier, HELP TEXT      01150000
|          DC    C'AP2SVPEX'        SHOW NAME                           01160000
|          DC    C'&SYSDATE'             CREATION                       01170000
|  TELLTEXT DS   0H                  START OF HELP                       01200000
|   TELL 'APL2/CMS Sample GSVP Installation Exit (Called from AP2CSVP)'  01210000
|   TELL ' '                                                             01220000
|   TELL 'Syntax: AP2SVPEX IDENTIFY'                                     01230000
|   TELL '  Arguments: R0 -> EPLIST + C -> VMCMHDR'                      01240000
|   TELL '     User       = VMCMUSER = VM userid to validate for GSVP'   01250000
|   TELL '     Account-ID = VMCMMID  = 0 or proposed APL account number' 01260000
|   TELL ' '                                                             01270000
|   TELL 'Operation:'                                                    01280000
|   TELL '  Does "STATE" for SVPEXIT EXEC'                               01290000
|   TELL '      If SVPEXIT EXEC does not exist,'                         01300000
|   TELL '        Then return as if AP2SVPEX did not exist.'             01310000
|   TELL '         (Authorizes user if account-ID ¬= 0)'                 01320000
|   TELL '        Else "EXEC SVPEXIT IDENTIFY User Account-ID"'          01330000
|   TELL '  Assign ID and authorize for GSVP using EXEC return code (RC):' 01340000
|   TELL '      If RC > 0    , assign RC as ID and authorize user'       01350000
|   TELL '      If RC < -1001, assign -RC as ID but do not authorize user' 01360000
|   TELL '         Otherwise , do not assign ID and do not authorize user' 01370000
|   TELL ' '                                                             01380000
|   TELL 'NOTES on SVPEXIT EXEC:'                                        01390000
|   TELL '   May use facilites like NAMEFIND to locate information.'      01400000
|   TELL '   May not use any functions dependent on external interrupts'  01410000
|   TELL '       (e.g. IUCV, VMCF, timers) because it is executing'      01420000
|   TELL '       as disabled exit during an external interrupt.'         01430000
|          DC    AL1(0)             END TELL                            01450000
|  LTELL   EQU   *-TELLTEXT         Total Length                        01460000
|          EJECT 1                                                       01470000
|  *                                                                     01480000
|  ***           ESTABLISH AS NUCLEUS EXTENSION                          01490000
|  *                                                                     01500000
|  BYID    SPKA  0                  SWITCH TO KEY ZERO                  01510000
|          LR    R8,R14              SAVE CMSRET FOR QUICK OUTS          01520000
```

```
|          CLI   WR1,X'FE'           EDGE CONDITION SERVICE CALL?        01530000
|          BNLR  R8                    - YES, QUIT NOW                   01540000
|          BAL   R14,C@NUCXQ           SEE IF ALREADY THERE              01550000
|          BNZ   INRELO                - NO, GO RELOCATE                 01560000
|  *                                 ALREADY NUCXLOADED                  01570000
|          C     R2,WR2              WE OUGHT TO BE IT                   01580000
|          BE    INALIAS               - AND WE ARE                      01590000
|          BCTR  R15,R8                - EXIT, RC = -1 (COULD HAPPEN)    01600000
|  *                                                                     01610000
|  ***           RELOCATE CODE AND DECLARE AS NUC EXTENSION              01620000
|  ***                    (NUCXDROP WILL FREE THE STORAGE)               01630000
|  *                                                                     01640000
|  INRELO   BCTR  R15,R8             MAKE SURE RC WAS 1 (NUCEXT EXISTS)  01650000
|          LA    R0,XRELODW          SIZE IN DWORDS                      01660000
|          DMSFREE DWORDS=(0),ERR=*                                      01670000
|          LTR   R15,R15             ANY STORAGE LEFT ?                  01680000
|          BNZR  R8                    - NO, RETURN TO SENDER            01690000
|          LR    R4,R1               YES, SAVE 'TO' ADDRESS              01700000
|          LR    R0,R1               COPY FOR DECLARE                    01710000
|          L     R2,XRELOL           AND GIVE LENGTH                     01720000
|          LA    R1,$NUCNAME         SET TASKNAME                        01730000
|          SLR   R15,R15                 NO FLAGS                        01740000
|          BAL   R14,C@NUCXA             DECLARE NUCX                    01750000
|          BZ    INRELOM             - IF IT WORKED, GO RELOCATE         01760000
|          LR    R2,R15              ELSE, SAVE RETURN  CODE             01770000
|          LA    R0,XRELODW          SIZE IN DWORDS                      01780000
|          LR    R1,R4                       LOCATION                    01790000
|          DMSFRET DWORDS=(0),LOC=(1),ERR=*                              01800000
|          LR    R15,R2              RESTORE RETURN CODE                 01810000
|          BR    R8                  - BACK TO CMS                       01820000
|  *                                                                     01830000
|  **            EXTENSION NAME DECLARED                                 01840000
|  *                                                                     01850000
|          USING SCBLOCK,R2                                              01860000
|  INRELOM  DS    0H                 RELOCATE CODE (R4 HAS 'TO')         01870000
|          L     R5,XRELOL                  LENGTH                      01880000
|          LR    R0,R12              'FROM' ADDRESS                      01890000
|          LR    R1,R5                      LENGTH                      01900000
|          MVCL  R4,R0               RELOCATE THIS CODE                  01910000
|          L     R12,SCBXORG         TAKE DELAYED BRANCH                 01920000
|          B     INALIAS             - RIGHT NOW                         01930000
|  *                                                                     01940000
|  ***           SET SCBNAME AND PSW                                     01950000
|  *                                                                     01960000
|  INALIAS  MVC   SCBNAME,$NUCNAME   INSURE RIGHT RUN-TIME NAME          01970000
|          XC    SCBPSW(2),SCBPSW           AND DISABLED KEY 0          01980000
|          TM    X'5EA'(0),X'80'     XA machine?                         01982000
|          BNO   *+10                - No, skip                          01984000
|          OC    SCBPSW(2),=X'0008'  Always on in XA PSW                 01986000
|  *                                                                     01990000
|  **       RELOCATION COMPLETE - REENTER BELOW AS IF INVOKED THERE       02000000
|  *                                                                     02010000
|          LM    R14,R1,WR14         RELOAD PARM REGISTERS               02020000
|          LA    R15,NUCSVPEX        SET COMMAND ENTRY                   02030000
```

```
|          ST    R15,SCBENTR          ADDRESS IN NEW CODE                02040000
|          LR    R12,R15              BASE REG IS GUARANTEED             02050000
|          LPSW  SCBPSW              - ENTER NUCSVPEX                    02060000
|          SPACE 1                                                       02070000
| XRELOL   DC    A(ZZSIZE)           LENGTH TO MOVE                      02080000
| XRELODW  EQU   ZZSIZE/8            DOUBLE-WORDS TO ACQUIRE             02090000
|          DROP  ,                                                       02100000
|          TITLE 'Command Handler'                                       02110000
| ********************************************************************** 02120000
| *                                                                    * 02130000
| * NUCSVPEX- COMMAND PARSER                                           * 02140000
| *                                                                    * 02150000
| * FORMAT  - AP2SVPEX FUNC                                            * 02160000
| *                                                                    * 02170000
| *                                                                    * 02180000
| ********************************************************************** 02190000
|          USING SCBLOCK,R2          MAP CMS NUCLEUS EXTENSION BLOCK     02200000
|          USING TKLIST,R9           MAP CMS TOKENIZE PLIST              02210000
|          USING WORK,R13            MAP CMS SVC SAVEARA (USAVE)         02220000
|          ENTRY NUCSVPEX                                                02230000
|          USING AP2SVPEX,R12                                            02240000
|          DS    0D                                                      02250000
| $NUCNAME DC    CL8'AP2SVPEX'       USE CONSTANT AS EYE-CATCHER         02260000
| NUCSVPEX DS    0D                  ALIGN ENTRY FOR BRANCHES            02270000
|          L     R12,SCBXORG         RELOCATE ADDRESSABILITY             02280000
|          LA    R15,ZRCOP           ASSUME 'INVALID OPERAND'            02290000
|          STM   R14,R2,WR14         SAVE REGISTERS                      02300000
|          SPACE 1                                                       02310000
|          LR    R9,R1               COPY TOKENIZED LIST                 02320000
|          LR    R1,R0               COPY EPLIST                         02330000
|          SLR   R3,R3               ASSUME NO EPLIST                    02340000
|          CLI   WR1,EPLFUNC         CALLED AS FUNCTION ?                02350000
|          BNE   DCMLOOK               - NO                              02360000
|          USING EPLIST,R1                                               02370000
|          L     R3,EPLOPT           OPTIONS                             02380000
|          DROP  R1                                                      02390000
| DCMLOOK  LA    R5,DISPCMD-DISPCMDL POINT AT COMMAND TABLE              02400000
|          LA    R4,DISPCMDL         LENGTH                              02410000
| DCMDLP   ALR   R5,R4               BUMP TO NEXT ENTRY                  02420000
|          CLC   TKLVERB,0(R5)       IS IT THE ONE ?                     02430000
|          BH    DCMDLP                - NO, LOOP                        02440000
|          BLR   R14                   - NO, AND NOT POSSIBLE            02450000
|          LH    R4,8(,R5)           YES, GET OFFSET                     02460000
|          BAL   R8,DISPCMD(R4)      - ENTER ROUTINE                     02470000
|          DROP  R9                                                      02480000
| DCMRET   L     R14,WR14            AND ACMSRET                         02490000
|          BR    R14                 BACK TO CMS                         02500000
|          SPACE 2                                                       02510000
| *                                                                      02520000
| *        COMMAND TABLE MUST BE IN EBCDIC COLLATING ORDER               02530000
| *                                                                      02540000
| DISPCMD  DS    0F                                                      02550000
|          DC    CL8'?       ',AL2(D$TELL-DISPCMD)                       02560000
| DISPCMDL EQU   *-DISPCMD                                               02570000
```

```
|          DC    CL8'IDENTIFY',AL2(D$ID-DISPCMD)                          02580000
|          DC    8X'FF',AL2(D$TELL-DISPCMD)         CATCH'EM ALL         02590000
|          TITLE 'Command Handler - IDENTIFY'                            02600000
| *-------------------------------------------------------------------* 02610000
| * ID     - Identify user for GSVP                                   * 02620000
| *                                                                   * 02630000
| *          Issue 'EXEC SVPEXIT IDENTIFY' VMCMUSER VMCMMID           * 02640000
| *                                                                   * 02650000
| *   INPUT                                                           * 02660000
| *        R3 = EPLOPT  points at VMCMHDR passed from AP2CSVP         * 02670000
| *        R8 = RETURN ADDRESS                                        * 02680000
| *        R9 = CMS TOKENIZED PLIST                                   * 02690000
| *                                                                   * 02700000
| *-------------------------------------------------------------------* 02710000
|          USING TKLIST,R9                                             02720000
| D$ID     DS    0H                                                    02730000
|          LTR   R3,R3               EPLOPT PRESENT ?                  02740000
|          BNPR  R8                    - NO, ERROR                     02750000
|          USING VMCMHDR,R3                                            02760000
|          CMD   $STATE                YES, IS EXEC FOUND?             02770000
|          L     R15,$MINUS1         ASSUME NOT (BUT LEAVE CC ALONE)   02780000
|          BNZR  R8                    - RETURN IF NOT                 02790000
|          MVC   WCMDCMD,$EXEC                                         02800000
|          MVC   WCMDNAME,$SVPEXIT                                     02810000
|          MVC   WCMDID,TKLVERB        COPY 'IDENTIFY'                 02820000
|          L     R0,VMCMMID            CONVERT ID NUMBER               02830000
|          CVD   R0,WCMDUSER                                           02840000
|          UNPK  WCMDNUM,WCMDUSER                                      02850000
|          OI    WCMDNUM+7,X'F0'                                       02860000
|          MVC   WCMDUSER,VMCMUSER     MOVE USERID                     02870000
|          MVC   WCMDFF,$FENCE                                         02880000
|          CMD   WCMD                                                  02890000
|          BNP   D$IDNO              - BRANCH IF EXEC REJECTS USER     02900000
|          ST    R15,VMCMMID         STORE REVISED ID                  02910000
|          SLR   R15,R15             CLEAR RETURN CODE                 02920000
|          BR    R8                  - RETURN                          02930000
| D$IDNO   DS    0H                  NEGATIVE RETURN CODE              02940000
|          LCR   R15,R15             GET MAGNITUDE                     02950000
|          C     R15,=A(1001)        COMPARE TO MINIMUM                02960000
|          BL    D$IDNO2             - TOO LOW, DON'T USE IT           02970000
|          ST    R15,VMCMMID         STORE REVISED ID                  02980000
| D$IDNO2  LA    R15,ZRCNO           TELL AP2CSVP ID NOT AUTHORIZED    02990000
|          BR    R8                  - BY RETURNING WITH R15=1         03000000
|          DROP  R3,R9                                                 03010000
|          TITLE 'Command Handler - ?'                                 03020000
| *-------------------------------------------------------------------* 03030000
| * ?      - Show HELP text                                           * 03040000
| *-------------------------------------------------------------------* 03050000
| D$TELL   DS    0H                                                    03060000
|          LA    R2,TELLTEXT         START OF DATA                     03070000
|          BALR  R14,0               LOOP HEAD                         03080000
|          CLI   0(R2),00            END OF DATA ?                     03090000
|          BE    D$TZ                - YES, ONWARD                     03100000
|          LINEDIT TEXTA=(R2),DOT=NO,COMP=NO,MF=(E,WLINEDIT)           03110000
```

```
|           LTR   R15,R15             STILL OK ?                          03120000
|           BNZ   D$TZ                  - NO                              03130000
|           IC    R15,0(,R2)            YES, GET LENGTH OF LAST           03140000
|           LA    R2,1(R15,R2)        BUMP TO NEXT                        03150000
|           BR    R14                 - LOOP                              03160000
|  D$TZ     LA    R15,ZRCTELL                                             03170000
|           BR    R8                  - EXIT WITH RC SET                  03180000
|           TITLE 'C@ Subroutines - Nucleus Extensions'                     03190000
|  *                                                                      03200000
|  * C@NUCXA      DECLARE A NUCEXT ALIAS                                  03210000
|  *                                                                      03220000
|  *              R0  = SCBXORG       R1  = WORK                          03230000
|  *              R1  = A(NUCXNAME)   R1  = WORK                          03240000
|  *              R2  = SCBXLEN       R2  = SCB                           03250000
|  *              R14 = RETURN                                            03260000
|  *              R15 = SCBSFLAG      R15 = RETURN CODE                   03270000
|  *                                                                      03280000
|  C@NUCXA  DS    0H                                                      03290000
|           MVC   WCMD(L'$NUCDCL),$NUCDCL NO, SET DECLARE                 03300000
|           ST    R0,WCMDORG          AND ORIGIN AS DEFAULT ENTRY POINT   03310000
|           ST    R0,WCMDPSW+4                                            03320000
|           ST    R2,WCMDLEN          SET LENGTH                          03330000
|           STC   R15,WCMDSFL         AND FLAGS                           03340000
|           LR    R0,R1               PRESERVE SERVER                     03350000
|           MVC   WCMDNAME,0(R1)          FOR THIS CMD                    03360000
|           CMD   WCMD                DECLARE NUCLEUS EXTENSION           03370000
|           BNZR  R14                 - RETURN ON FAILURE                 03380000
|           LR    R1,R0               RESTORE SERVER                      03390000
|           B     C@NUCXQ             - DO QUERY FOR SCB                  03400000
|           SPACE 3                                                       03410000
|  *                                                                      03420000
|  * C@NUCXQ      DO NUCEXT OR SUBCOM QUERY                               03430000
|  *                                                                      03440000
|  *              R0  = WORK                                              03450000
|  *              R1  = A(NUCXNAME)   R1  = WORK                          03460000
|  *              R2  = WORK          R2  = SCB                           03470000
|  *              R14 = RETURN                                            03480000
|  *              R15 = WORK          R15 = RETURN CODE                   03490000
|  *                                                                      03500000
|  C@NUCXQ  DS    0H                                                      03510000
|           MVC   WCMD(L'$NUCQ),$NUCQ SET NUCEXT QUERY                   03520000
|           MVC   WCMDNAME,0(R1)          AND THIS SERVER                 03530000
|           CMD   WCMD                ASK IF ALREADY NUCEXT               03540000
|           L     R2,WCMDSCB                                              03550000
|           BR    R14                 -RETURN (CC IS SET)                 03560000
|           SPACE 3                                                       03570000
|           TITLE 'Constants and DSECTs'                                  03580000
|           DROP  ,                                                       03590000
|  CMS202   EQU   202                 MARK IN XREF WHERE CMS CALLED       03600000
|  ZRCNO    EQU   1                   USER NOT AUTHORIZED BUT ID SET      03610000
|  ZRCOP    EQU   24                  INVALID OPERAND                     03620000
|  ZRCTELL  EQU   100                 HELP TEXT PRINTED                   03630000
|  *                                                                      03640000
|  *        CONSTANTS                                                     03650000
```

```
|  *                                                                    03660000
|  *                                                                    03670000
|           DS    0D                                                   03680000
|  $NUCDCL  DS    0CL40               MODEL FOR DECLARE                03690000
|  $NUCQ    DS    0CL24                     FOR QUERY                  03700000
|  $NUCEXT  DC    CL8'NUCEXT'                                          03710000
|  $BLANKS  DC    CL8' '                                               03720000
|  $ZERO    DC    H'0',AL1(0),AL1(0)  PSW, NUCEXT FLAGS, USER FLAGS    03730000
|  $MINUS1  DC    F'-1'               QUERY INDICATOR                  03740000
|           DC    3F'0'               USER WORD, ORIGIN, LENGTH        03750000
|  $FENCE   DC    8X'FF'                                               03760000
|  $STATE   DC    CL8'STATE'                                           03770000
|  $SVPEXIT DC    CL8'SVPEXIT'                                         03780000
|  $EXEC    DC    CL8'EXEC '                                           03790000
|           DC    CL8'*    '                                           03800000
|           DC    8X'FF'                                               03810000
|  *        END OF NUCXDCL PLIST                                       03820000
|           LTORG ,                                                    03830000
|  *                                                                   03840000
|  ***      END OF RELOCATED CODE                                      03850000
|  *                                                                   03860000
|           DS    0D                  FORCE ALIGNMENT                  03870000
|  ZZPROG   EQU   *                                                    03880000
|  ZZSIZE   EQU   ZZPROG-AP2SVPEX                                      03890000
|           EJECT 1                                                    03900000
|  WORK     DSECT ,                   MAP USAVE                        03910000
|           DS    F                                                    03920000
|  WR14     DS    F                   SAVED ENTRY REGS                 03930000
|  WR15     DS    F                                                    03940000
|  WR0      DS    F                                                    03950000
|  WR1      DS    F                                                    03960000
|  WR2      DS    F                                                    03970000
|  WTEMP    DS    6D                  SCRATCH IN-LINE USE ONLY         03980000
|           ORG   WTEMP                                                03990000
|  WCMD     DS    0CL48               COMMAND BUILD AREA               04000000
|  WCMDCMD  DS    CL8                 'SUBCOM' OR 'NUCEXT'             04010000
|  WCMDNAME DS    CL8                 TARGET COMMAND                   04020000
|  WCMDSCB  DS    0F                  SCB ADDRESS RETURNED FROM QUERY  04030000
|  WCMDPSW  DS    D                   PSW                              04040000
|  WCMDSFL  EQU   WCMDPSW+2,1         NUCEXT FLAGS                     04050000
|  WCMDUFL  EQU   WCMDPSW+3,1         USER   FLAGS                     04060000
|           DC    A(0)                USER WORD                        04070000
|  WCMDORG  DC    A(*-*)              ORIGIN  (USED BY DMSABN)          04080000
|  WCMDLEN  DC    A(*-*)              LENGTH  (USED BY DMSABN)          04090000
|           ORG   WCMDSCB                                              04100000
|  WCMDID   DC    CL8'IDENTIFY'       FUNCTION                         04110000
|  WCMDUSER DS    CL8                 USERID                           04120000
|  WCMDNUM  DS    CL8                 NUMBER                           04130000
|  WCMDFF   DC    8X'FF'              FENCE                            04140000
|           ORG   WTEMP                                                04150000
|  WLINEDIT LINEDIT DOT=NO,COMP=NO,MF=L HELP TEXT MESSAGE LIST         04160000
|           ORG                                                        04170000
|  WORKL    EQU   *-WORK                                               04180000
```

```
|          EJECT 1                                                    04190000
|  EPLIST  DSECT ,                 EXTENDED COMMAND PLIST             04200000
|  EPLCMD  DS    A                 COMMAND NAME BEGINNING             04210000
|  EPLARG  DS    A                 ARGUMENT START                     04220000
|  EPLARGZ DS    A                           END                      04230000
|  EPLOPT  DS    A                 OPTIONAL PARAMETER OR ZERO         04240000
|  EPLL    EQU   *-EPLIST                                            04250000
|          SPACE 1                                                    04260000
|  EPLFUNC EQU   1                                                    04270000
|          SPACE 2                                                    04280000
|  TKLIST  DSECT ,                 TOKENIZED PLIST                    04290000
|          DS    CL8                                                  04300000
|  TKLVERB DS    CL8               COMMAND VERB                       04310000
|  TKLL    EQU   *-TKLIST                                            04320000
|          SPACE 1                                                    04330000
|          TITLE 'VM Data Areas'                                      04340000
|          SCBLOCK ,               NUCEXT INTERFACE                   04350000
|          REGEQU ,                REGISTERS                          04360000
|          COPY VMCBLOKS           VMCF (FROM DMKSP MACLIB)           04370000
|          END   ,                                                    04380000
```

TCP/IP Profile File — AP2TCPIP APL2PROF

```
| ************************************************************************
| *
| *  APL2 Version 2 Release 2
| *  Licensed Materials - Property of IBM
| *  5688-228, 5688-229 (C) COPYRIGHT IBM CORP. 1992, 1994.
| *  See Copyright Instructions
| *
| ************************************************************************
| *
| * Sample APL2 TCP/IP Profile File
| *
| * In the VM/CMS environment, the TCP/IP profile file is a sequential
| * file called AP2TCPIP APL2PROF and must be on an accessed CMS minidisk.
| * In the MVS/TSO environment, the TCP/IP profile file is a member of a
| * partitioned data set.  The member name is AP2TCPIP.  The partitioned
| * data set must be allocated to ddname APL2PROF.  Concatenated
| * allocation is supported and can be used to override profile files.
| *
| * The TCP/IP profile file may contain fixed or variable length records
| * with a maximum record length of 255 bytes.  Any line that starts with
| * an * is a comment.

| * There are 2 types of entries in a TCPIP profile:
| *
| * Identification entry (:svopid):
| *   These entries are used to convert the left argument of □SVO
| *   to an extended ID (xid). They are also used to convert an xid
| *   from a remote processor back to a single processor number.
| *
| *   ID entries are required to make an offer to a local processor
| *   dependent on a parent different from the offerer or to make an
| *   offer to a processor on a different machine and/or userid.
| *
| * Authorization entry (:procauth):
| *   These entries are used to authorize remote processors to share
| *   variables with local processors.
| *
| * Any line that starts with an * is a comment.
| *
| * Each data line in a TCP/IP profile file can contain one or more tags
| * and data.  Tag data may not be continued onto new records.  Each entry
| * must begin with an :svopid tag or a :procauth tag. For example:
| *
| *      +-----------------------------------------+
| *      |                                         |
| *      |   * Sample remote processor identification |
| *      |   :svopid.34567                         |
| *      |      :address.123.45.234.64             |
| *      |      :userid.johndoe                    |
| *      |      :processor.127,23435               |
| *      |                                         |
```

```
| *      |   * Sample local processor identification  |
| *      |   :svopid.34567                            |
| *      |      :processor.127,1002                   |
| *      |                                            |
| *      |   * Sample authorization                   |
| *      |   :procauth.127,23435                      |
| *      |      :rsvopid.34567                        |
| *      |                                            |
| *      +--------------------------------------------+
| *
| *
| * TCP/IP PROFILE TAGS
| *
| * The format of TCP/IP profile tags is as follows:
| *
| * :tag.value
| *
| * where :tag. is chosen from the following set of keywords and
| * identifies the meaning of value.  Tags and their values can be coded
| * in either upper, lower, or mixed case letters.  There are no column
| * dependencies except that asterisks identifying comment records must
| * appear in column 1.  When an :svopid or :procauth tag occurs multiple
| * times in a TCP/IP profile entry, only the first occurrence of the tag
| * is used.
| *
| * :svopid.id    (numeric - 0 to 32767)
| *
| *          The processor ID number specified by the local user in the
| *          left argument of □SVO or the number from which □SVQ
| *          will report offers have arrived.  This tag identifies the
| *          beginning of an identification entry and is required.
| *
| *
| * :address.addr  (4 part dotted numeric 1-255 or 0)
| *
| *          The IP address of the potential partner's machine in standard
| *          internet format consisting of four decimal numbers between 0
| *          and 255 separated by decimal points.  This tag is optional.
| *          The default is the local IP address or 0 if TCP/IP is not
| *          installed.
| *

| * :userid.user   (alphanumeric)
| *
| *          The userid of a potential partner with whom you may share
| *          variables. The value must be coded in the correct case.
| *          This tag is optional. The default is the local userid
| *          or 0 if TCP/IP is not installed.
| *
| * :processor.id  (1 to 3 numeric separated by commas)
| *
| *          A processor number and optionally a parent and pparent.
| *
| * :procauth.id   (1 to 3 numeric - 1 to 32767 separated by commas or '*')
```

```
| *
| *          This entry authorizes one or more remote processors
| *          to share with a local processor. The id value specifies
| *          the local processor to which shares are being authorized.
| *          An '*' defines authorization for ANY local processor.
| *          This tag identifies the beginning of an authorization
| *          entry and is required.
| *
| *
| * :rsvopid.id   (1 or more numeric separated by commas or '*')
| *
| *          This entry identifies the processors which are authorized
| *          to share with the processor identified by :procauth. Each
| *          number refers to an :svopid entry identifying the authorized
| *          remote processor. An '*' authorizes all processors.
| *
| *
| * :tcpipsrv.id
| *
| *          The port number being used by the TCP/IP port server on the
| *          potential partner's machine.  This tag is optional; it
| *          defaults to 31415.
| *
| *
| *
| * APL2 Version 2 Release 1 used a different format for the profile.
| * This format is still accepted but may be used only when sharing
| * with Version 2 Release 1. See the Release 1 documentation for a
| * description of the format.
| *
| *      +---------------------------------------+
| *      |                                       |
| *      |   * Sample TCP/IP entry for id 34567  |
| *      |   * Version 2 Release 1 Compatibility |
| *      |   :SVOPID.34567                       |
| *      |      :PROTOCOL.TCPIP                  |
| *      |      :RADDRESS.123.45.234.64          |
| *      |      :USERID.JOHNDOE                  |
| *      |      :PROCID.127                      |
| *      |                                       |
| *      +---------------------------------------+
| *
```

AP2VIOPE ASSEMBLE Sample — AP2VIOPE AP2MSAMP

```
VIOPE    TITLE 'P-AP2VIOPE: INSTALLATION OPTIONS FOR APL2AE/CMS'        00014990
************************************************************          00020000
*                                                          *          00030000
* MODULE NAME = AP2VIOPE                                   *          00040000
*                                                          *          00050000
* DESCRIPTIVE NAME = APL2AE INSTALLATION OPTIONS FOR CMS    *          00060000
*                                                          *          00070000
* COPYRIGHT:  Licensed Materials - Property of IBM         *          00086990
*                                                          *          00093980
*             5688-228, 5688-229                           *          00104960
*             (C) COPYRIGHT IBM CORP. 1987, 1994           *          00111940
*                                                          *          00114950
*             See Copyright Instructions                   *          00121940
*                                                          *          00128930
* STATUS:     APL2 VERSION 2 RELEASE 2                     *          00142910
*                                                          *          00150000
* FUNCTION = APL2 INVOCATION OPTIONS FOR CMS.              *          00160000
*                                                          *          00170000
* CHANGE ACTIVITY = NONE                                   *          00180000
*                                                          *          00190000
* MACROS = AP2VOPT                                         *          00200000
*                                                          *          00210000
************************************************************          00220000
          SPACE 1                                                     00230000
AP2VIOPE CSECT                                                        00240000
          ENTRY AP2VIOPT                                              00250000
AP2VIOPT EQU   *                                                      00260000
          AP2VOPT                                                    *00270000
               DEFAULT=(DEFAULTA,DEFAULTZ),   DEFAULT PARAM LIST     *00290000
               OVERIDE=(OVERRIDA,OVERRIDZ),  OVERRIDE PARAM LIST     *00300000
               ATASKS=,                                              *00310990
               RESEPS=(AP2VN10,    REXX RESOLVER                     *00311980
               AP2VN11,            MODULE RESOLVER (FORTRAN, ETC.)   *00312970
               AP2VN12,            FILES AS VARIABLES                *00313960
               AP2X119,            TCP/IP AP                         *00317990
               AP2X127,            SQL/DS AP                         *00318980
               AP2X211,            APL OBJECTS AP                    *00320000
               AP2XNRSS),          VARIOUS EXTERNAL FUNCTIONS        *00331980
               RESAPS=(AP2V100,    COMMAND PROCESSOR                 *00340000
               AP2V101,            STACK PROCESSOR                   *00350000
               AP2V102,            MAIN STORAGE DISPLAY              *00360000
               AP2V110,            CMS FILE AP                       *00370000
               AP2V111,            QSAM FILE AP                      *00380000
               AP2V123,            VSAM FILE AP                      *00390000
               AP2X121,            APL FILE AP                       *00400000
               AP2X124,            FULLSCREEN AP                     *00405000
               AP2X126),           GDDM AP                           *00410000
               CSVPID=AP2SVP,                GLOBAL SVP USERID       *00420000
               AP2SHNM=(AP2V22E1,AP2V22E2), LOW SAVED SEGMENT NAMES   *00437960
               AP2HSNM=(AP2V22F1,AP2V22F2), HIGH SAVED SEGMENT NAMES  *00438950
```

```
|               QNLT=ENU           PUT INITIAL NATIONAL LANGUAGE HERE   00441380
| *                                OR 'QNLT=' FOR UPPER CASE ENGLISH    00441980
|          SPACE 2                                                      00442970
|          ORG   OPTSVPID          BACK TO DEFINITION OF CSVPID         00443960
|          ENTRY AP2SVPID          GIVE IT AN ENTRY POINT               00444950
| AP2SVPID DS    0X                TO BE FOUND BY AP2VAPV2              00445940
|          ORG   ,                                                      00446930
|          TITLE 'AP2VIOPE - DEFAULT OPTIONS SETTINGS'                  00454990
| *---------------------------------------------------------------*     00460000
| *        THESE ARE THE DEFAULT SETTINGS FOR THE APL2            *     00474990
| *        INVOCATION OPTIONS.  THEY CAN BE OVERRIDDEN            *     00479980
| *        BY THE USER WHEN INVOKING APL2.                        *     00484970
| *                                                               *     00489960
| *        THE INVOCATION OPTIONS ARE DESCRIBED IN                *     00494950
| *        APL2 PROGRAMMING: SYSTEM SERVICES REFERENCE.           *     00499940
| *---------------------------------------------------------------*     00510000
| DEFAULTA DS    0H                                                     00520000
|          DC    C'APNAMES('                                            00543970
|          DC      C'AP2X127(ISOL(RR)),'                                00550960
|          DC      C'AP2X119(SERVPORT(31415) TCPID(TCPIP)),'             00561320
|          DC      C'AP2X211)'                                          00561930
|          DC    C'CASE(1)'         DEFAULT LOWER CASE, UNDERBARS OK    00564940
|          DC    C'DATEFORM(ISO)'   DATE FORMAT (US|EUROPE|ISO)         00571930
|          DC    C'DBCS(TRY)'       DOUBLE-BYTE CHARACTERS (TRY|ON|OFF|N) 00578920
|          DC    C'DEBUG(0)'        NO DEBUG FLAGS                      00585910
|          DC    C'DSOPEN(*)'       DSOPEN DEVICE TOKEN                 00592900
| *        DC    C'EXCLUDE()'       DO NOT START NAMED AUX PROCESSORS   00599890
|          DC    C'FREESIZE(64K)'   REQUIRED FREESIZE                   00606880
|          DC    C'ID(0)'           USER ID                             00613870
| *        DC    C'INPUT('')TIME'')'  INITIAL INPUT LINES               00630000
| *        DC    C'QUIET(ON)'       NO OUTPUT UNTIL FIRST INPUT         00649990
| *        DC    C'RUN(CMSIVP)'     INITIAL PROGRAM TO RUN              00659980
|          DC    C'SHRSIZE(40K)'    SHARED MEMORY SIZE                  00673960
|          DC    C'SVMAX(88)'       MAXIMUM SIMULTANEOUS SHARED VARS    00679960
|          DC    C'SYSDEBUG(-255)'  TURN OFF ALL SYSDEBUG FLAGS         00700000
|          DC    C'TRACE(-65535)'   TURN OFF ALL TRACE FLAGS            00715990
|          DC    C'WSSIZE(25%)'     WORKSPACE SIZE                      00721980
| DEFAULTZ EQU   *-1                                                    00730000
|          TITLE 'AP2VIOPE - OVERRIDING OPTIONS SETTINGS'               00744990
| *---------------------------------------------------------------*     00750000
| *        THESE SETTINGS OVERRIDE THE INVOCATION PARAMETERS OF THE   *  00760000
| *        APL COMMAND.  THEY ARE CURRENTLY ALL COMMENTED OUT.        *  00770000
| *---------------------------------------------------------------*     00780000
| OVERRIDA DS    0H                                                     00790000
| *        DC    C'APNAMES(U(A)) '  LIST OF USER APNAMES AND PARAMETERS 00800000
| *        DC    C'CASE(1)       '  CASE                               00810000
| *        DC    C'DATEFORM(US)  '  DATE FORMAT (US|EUROPE|ISO)        00820000
| *        DC    C'DBCS(OFF)     '  DOUBLE-BYTE CHARACTERS (TRY|ON|OFF|N) 00830000
| *        DC    C'DEBUG(3)      '  FORCE ON SOME DEBUG FLAGS           00840000
| *        DC    C'DSOPEN()      '  DSOPEN DEVICE TOKEN                 00850000
| *        DC    C'EXCLUDE()     '  DO NOT START NAMED AUX PROCESSORS   00860000
| *        DC    C'FREESIZE(128K)'  DO NOT START UNLESS THIS AMT AVAIL  00870000
| *        DC    C'HILIGHT(INPUT)'  HILIGHTING (INPUT|OUTPUT|ON|OFF)   00880000
| *        DC    C'ID(0)         '  USER ID                             00890000
```

```
|  *        DC    C'INPUT('')LIB'')'  INITIAL INPUT LINES                00900000
|  *        DC    C'QUIET(ON)     '  NO OUTPUT UNTIL FIRST INPUT         00919990
|  *        DC    C'RUN(CMSIVP)   '  INITIAL PROGRAM TO RUN              00929980
|  *        DC    C'SHRSIZE(48K)  '  SHARED MEMORY SIZE                  00940000
|  *        DC    C'SVMAX(32)     '  MAXIMUM SIMULTANEOUS SHARED VARS    00960000
|  *        DC    C'SYSDEBUG(-255)'  TURN OFF ALL SYSDEBUG FLAGS         00970000
|  *        DC    C'TRACE(-65535) '  TURN OFF ALL TRACE FLAGS           00980000
|  *        DC    C'WSSIZE(600K)  '  WORKSPACE SIZE                     00990000
|  OVERRIDZ EQU   *-1                                                   01000000
|           END                                                         01010000
```

AP2VIOPT ASSEMBLE Sample — AP2VIOPT AP2MSAMP

```
VIOPT    TITLE 'P-AP2VIOPT: INSTALLATION OPTIONS FOR APL2/CMS'          00014990
************************************************************          00020000
*                                                          *          00030000
* MODULE NAME = AP2VIOPT                                   *          00040000
*                                                          *          00050000
* DESCRIPTIVE NAME = APL2 INSTALLATION OPTIONS FOR CMS     *          00060000
*                                                          *          00070000
* COPYRIGHT:  Licensed Materials - Property of IBM         *          00086990
*                                                          *          00093980
*             5688-228                                     *          00104960
*             (C) COPYRIGHT IBM CORP. 1984, 1994           *          00112930
*                                                          *          00114950
*             See Copyright Instructions                   *          00121940
*                                                          *          00128930
* STATUS:     APL2 VERSION 2 RELEASE 2                     *          00145900
*                                                          *          00150000
* FUNCTION = APL2 INVOCATION OPTIONS FOR CMS.              *          00160000
*                                                          *          00170000
* CHANGE ACTIVITY = NONE                                   *          00180000
*                                                          *          00190000
* MACROS = AP2VOPT                                         *          00200000
*                                                          *          00210000
************************************************************          00220000
          SPACE 1                                                     00230000
AP2VIOPT CSECT                                                        00240000
          AP2VOPT                                                    *00250000
                DEFAULT=(DEFAULTA,DEFAULTZ),   DEFAULT PARAM LIST    *00260000
                OVERIDE=(OVERRIDA,OVERRIDZ),  OVERRIDE PARAM LIST    *00270000
                ATASKS=,                                             *00272000
                RESEPS=(AP2VN10,    REXX RESOLVER                    *00274860
                AP2VN11,            MODULE RESOLVER (FORTRAN, ETC.)  *00274950
                AP2VN12,            FILES AS VARIABLES               *00275040
                AP2X104,            )COPY AP                         *00275470
                AP2X119,            TCP/IP AP                        *00275770
                AP2X127,            SQL/DS AP                        *00276170
                AP2X211,            APL OBJECTS AP                   *00276570
                AP2IPDD2,           EDIT/IN/OUT                      *00278570
                AP2XNRSS),          VARIOUS EXTERNAL FUNCTIONS       *00281040
                RESAPS=(AP2V100,    COMMAND PROCESSOR                *00284990
                AP2V101,            STACK PROCESSOR                  *00290000
                AP2V102,            MAIN STORAGE DISPLAY             *00300000
                AP2V110,            CMS FILE AP                      *00310000
                AP2V111,            QSAM FILE AP                     *00320000
                AP2V123,            VSAM FILE AP                     *00330000
                AP2X124,            FULLSCREEN AP                    *00336990
                AP2X120,            SESSION MANAGER AP               *00340000
                AP2X121,            APL FILE AP                      *00350000
                AP2X126),           GDDM AP                          *00360000
                CSVPID=AP2SVP,                GLOBAL SVP USERID      *00370000
                AP2SHNM=(AP2V22V1,AP2V22V2), LOW SHARED SEGMENT NAMES *00389740
                AP2HSNM=(AP2V22W1,AP2V22W2), HIGH SHARED SEGMENT NAME *00390130
```

```
|               QNLT=ENU           PUT INITIAL NATIONAL LANGUAGE HERE   00390580
| *                                OR 'QNLT=' FOR UPPER CASE ENGLISH    00390670
|          SPACE 2                                                      00391000
|          ORG   OPTSVPID          BACK TO DEFINITION OF CSVPID         00392000
|          ENTRY AP2SVPID          GIVE IT AN ENTRY POINT               00393000
| AP2SVPID DS    0X                TO BE FOUND BY AP2VAPV2              00394000
|          ORG   ,                                                      00395000
|          TITLE 'AP2VIOPT - DEFAULT OPTIONS SETTINGS'                  00404990
| *---------------------------------------------------------------*     00410000
| *        THESE ARE THE DEFAULT SETTINGS FOR THE APL2            *     00424990
| *        INVOCATION OPTIONS.  THEY CAN BE OVERRIDDEN            *     00429980
| *        BY THE USER WHEN INVOKING APL2.                        *     00434970
| *                                                               *     00439960
| *        THE INVOCATION OPTIONS ARE DESCRIBED IN                *     00444950
| *        APL2 PROGRAMMING: SYSTEM SERVICES REFERENCE.           *     00449940
| *---------------------------------------------------------------*     00460000
| DEFAULTA DS    0H                                                     00470000
|          DC    C'APNAMES('                                            00489960
|          DC      C'AP2X104,'                                          00493950
|          DC      C'AP2X119(SERVPORT(31415) TCPID(TCPIP)),'             00495940
|          DC      C'AP2X127(ISOL(RR)),'                                00497940
|          DC      C'AP2X211)'                                          00501930
|          DC    C'CASE(1)'         DEFAULT LOWER CASE, UNDERBARS OK    00505920
|          DC    C'DATEFORM(ISO)'   DATE FORMAT (US|EUROPE|ISO)         00509910
|          DC    C'DBCS(TRY)'       DOUBLE-BYTE CHARACTERS (TRY|ON|OFF|N) 00513900
|          DC    C'DEBUG(0)'        NO DEBUG FLAGS                      00517890
|          DC    C'DSOPEN(*)'       DSOPEN DEVICE TOKEN                 00521880
| *        DC    C'EXCLUDE()'       DO NOT START NAMED AUX PROCESSORS   00525870
|          DC    C'FREESIZE(256K)'  REQUIRED FREESIZE                   00534990
|          DC    C'HILIGHT(INPUT)'  HILIGHTING (INPUT|OUTPUT|ON|OFF)   00540000
|          DC    C'ID(0)'           USER ID                             00554990
| *        DC    C'INPUT('')TIME'')'  INITIAL INPUT LINES               00560000
|          DC    C'PROFILE(DEFAULT)'  SESSION MANAGER PROFILE FILE      00570000
| *        DC    C'QUIET(ON)'       NO OUTPUT UNTIL FIRST INPUT         00585990
| *        DC    C'RUN(CMSIVP)'     INITIAL PROGRAM TO RUN              00591980
|          DC    C'SHRSIZE(40K)'    SHARED MEMORY SIZE                  00599960
|          DC    C'SMAPL(TRY)'      SESSION MANAGER (TRY|ON|OFF)        00603960
|          DC    C'SVMAX(88)'       MAXIMUM SIMULTANEOUS SHARED VARS    00609950
|          DC    C'SYSDEBUG(-255)'  TURN OFF ALL SYSDEBUG FLAGS         00620000
|          DC    C'TRACE(-65535)'   TURN OFF ALL TRACE FLAGS            00635990
|          DC    C'WSSIZE(25%)'     WORKSPACE SIZE                      00641980
| DEFAULTZ EQU   *-1                                                    00650000
|          TITLE 'AP2VIOPT - OVERRIDING OPTIONS SETTINGS'               00664990
| *---------------------------------------------------------------*     00670000
| *        THESE SETTINGS OVERRIDE THE INVOCATION PARAMETERS OF THE   *  00680000
| *        APL COMMAND.  THEY ARE CURRENTLY ALL COMMENTED OUT.        *  00690000
| *---------------------------------------------------------------*     00700000
| OVERRIDA DS    0H                                                     00710000
| *        DC    C'APNAMES(U(A)) '  LIST OF USER APNAMES AND PARAMETERS 00720000
| *        DC    C'CASE(1)       '  CASE                               00725000
| *        DC    C'DATEFORM(US)  '  DATE FORMAT (US|EUROPE|ISO)        00730000
| *        DC    C'DBCS(OFF)     '  DOUBLE-BYTE CHARACTERS (TRY|ON|OFF|N) 00735000
| *        DC    C'DEBUG(3)      '  FORCE ON SOME DEBUG FLAGS          00740000
| *        DC    C'DSOPEN()      '  DSOPEN DEVICE TOKEN                00750000
```

```
|  *         DC    C'EXCLUDE()     '  DO NOT START NAMED AUX PROCESSORS     00760000
|  *         DC    C'FREESIZE(128K)'  DO NOT START UNLESS THIS AMT AVAIL    00770000
|  *         DC    C'HILIGHT(INPUT)'  HILIGHTING (INPUT|OUTPUT|ON|OFF)      00780000
|  *         DC    C'ID(0)         '  USER ID                               00790000
|  *         DC    C'INPUT('')LIB'')'  INITIAL INPUT LINES                  00800000
|  *         DC    C'PROFILE(DEFAULT)'  SESSION MANAGER PROFILE DATASET     00810000
|  *         DC    C'QUIET(ON)     '  NO OUTPUT UNTIL FIRST INPUT           00822990
|  *         DC    C'RUN(CMSIVP)   '  INTIAL PROGRAM TO RUN                 00825980
|  *         DC    C'SHRSIZE(48K)  '  SHARED MEMORY SIZE                    00830000
|  *         DC    C'SMAPL(TRY)    '  SESSION MANAGER (TRY|ON|OFF)          00846990
|  *         DC    C'SVMAX(32)     '  MAXIMUM SIMULTANEOUS SHARED VARS      00854990
|  *         DC    C'SYSDEBUG(-255)'  TURN OFF ALL SYSDEBUG FLAGS           00860000
|  *         DC    C'TRACE(-65535) '  TURN OFF ALL TRACE FLAGS              00870000
|  *         DC    C'WSSIZE(600K)  '  WORKSPACE SIZE                       00880000
|  OVERRIDZ EQU   *-1                                                      00890000
|           END                                                            00900000
```

AP2XAPIC Module Sample — AP2XAPIC AP2MSAMP

```
| AP2XAPIC TITLE 'P-AP2XAPIC: APL2 Initialization Command for APL2PI'     00010000
| *-P-AP2XAPIC-----------------------------------------------*         00020000
| *                                                         *         00030000
| * MODULE-NAME: AP2XAPIC                                   *         00040000
| *                                                         *         00050000
| * DESCRIPTIVE-NAME: APL2 Initialization Command for APL2PI *         00060000
| *                                                         *         00070000
| * COPYRIGHT:   Licensed Materials - Property of IBM       *         00090980
| *                                                         *         00095980
| *              5688-228, 5688-229                         *         00111950
| *              (C) COPYRIGHT IBM CORP. 1988, 1994         *         00114940
| *                                                         *         00119950
| *              See Copyright Instructions                 *         00133930
| *                                                         *         00140000
| * STATUS:      APL2 VERSION 2 RELEASE 2                   *         00157970
| *                                                         *         00160000
| * FUNCTION:    Non-default APL2 initialization command used *         00170000
| *              by APL2PI.                                 *         00180000
| *                                                         *         00190000
| * NOTES:                                                  *         00200000
| *                                                         *         00210000
| *   DEPENDENCIES:                                        *         00220000
| *              Designed to be used in conjunction with APL2PI *       00230000
| *              (AP2VAPI or AP2TAPI).                      *         00240000
| *                                                         *         00250000
| *   RESTRICTIONS:                                        *         00260000
| *              None.                                      *         00270000
| *                                                         *         00280000
| *   REGISTER-CONVENTIONS:                                *         00290000
| *              None.                                      *         00300000
| *                                                         *         00310000
| *   PATCH-LABEL: None.                                   *         00320000
| *                                                         *         00330000
| * MODULE-TYPE: Table.                                     *         00340000
| *                                                         *         00350000
| *   PROCESSOR: ASSEMBLER H                               *         00360000
| *                                                         *         00370000
| *   MODULE SIZE: .1K (Design point, not current size)    *         00380000
| *                                                         *         00390000
| *   ATTRIBUTES:  REENTRANT, READ-ONLY.                   *         00400000
| *                                                         *         00410000
| *---------------------------------------------------------*         00420000
| *                                                         *         00430000
| * ENTRY-POINT: AP2XAPIC                                   *         00440000
| *                                                         *         00450000
| *   PURPOSE:   APL2 initialization command for APL2PI.   *         00460000
| *              SEE ALSO 'FUNCTION' IN THE ENTRY POINT PROLOG. *     00470000
| *                                                         *         00480000
| *   LINKAGE:   SEE 'CALL'/'ARGUMENTS' IN ENTRY POINT PROLOG *       00490000
| *                                                         *         00500000
| *---------------------------------------------------------*         00510000
```

```
|  *                                                               *         00520000
|  * INPUT:       SEE 'GLOBALS' IN THE ENTRY POINT PROLOGS       *         00530000
|  *                                                               *         00540000
|  * OUTPUT:      SEE 'GLOBALS' IN THE ENTRY POINT PROLOGS       *         00550000
|  *                                                               *         00560000
|  * EXIT-NORMAL: SEE 'RESULTS' IN THE ENTRY POINT PROLOGS       *         00570000
|  *                                                               *         00580000
|  * EXIT-ERROR:  SEE 'RESULTS' IN THE ENTRY POINT PROLOGS       *         00590000
|  *                                                               *         00600000
|  * EXTERNAL-REFERENCES:                                         *         00610000
|  *                                                               *         00620000
|  *   ROUTINES:  None.                                           *         00630000
|  *                                                               *         00640000
|  *   DATA-AREAS: None.                                          *         00650000
|  *                                                               *         00660000
|  *   CONTROL-BLOCKS: SEE LIST WHICH FOLLOWS THE ASSEMBLY        *         00670000
|  *                                                               *         00680000
|  * TABLES:      None.                                          *         00690000
|  *                                                               *         00700000
|  * MACROS:      SEE LIST WHICH FOLLOWS THE ASSEMBLY            *         00710000
|  *                                                               *         00720000
|  * CHANGE-ACTIVITY:                                            *         00730000
|  *                                                               *         00740000
|  *                                                               *         00750000
|  *                                                               *         00760000
|  *                                                               *         00770000
|  *                                                               *         00780000
|  *                                                               *         00790000
|  *-Z-------------------------------------------------------------*         00800000
|           TITLE 'A-AP2XAPIC: APL2 Initialization Command for APL2PI'      00810000
|  *-A-AP2XAPIC----------------------------------------------------*         00820000
|  *                                                               *         00830000
|  * ROUTINE:     AP2XAPIC                                       *         00840000
|  *                                                               *         00850000
|  * FUNCTION:    Optional APL2 initialization command used by   *         00860000
|  *              APL2PI (AP2VAPI or AP2TAPI).                   *         00870000
|  *                                                               *         00880000
|  * CALL:        None.                                          *         00890000
|  *                                                               *         00900000
|  * ARGUMENTS:   None.                                          *         00910000
|  *                                                               *         00920000
|  * RESULTS:     None.                                          *         00930000
|  *                                                               *         00940000
|  * DESCRIPTION: This module contains an APL2 initialization    *         00950000
|  *              command designed to be used with APL2PI.       *         00960000
|  *              This module may be modified to tailor the      *         00970000
|  *              command to an installation's or user's         *         00980000
|  *              requirements.   If this module is assembled    *         00990000
|  *              and link edited with APL2PI (ie: AP2VAPI or    *         01000000
|  *              AP2TAPI), APL2PI will use the command in this  *         01010000
|  *              module to invoke APL2.  If this module is      *         01020000
|  *              not link edited with APL2PI, APL2PI will use   *         01030000
|  *              a default command to invoke APL2.              *         01040000
```

```
*                                                               *         01050000
*-Z-------------------------------------------------------------*         01060000
         SPACE 1                                                          01070000
         ENTRY AP2XAPIC                                                   01080000
AP2XAPIC CSECT                                                            01090000
APL2CMD  DS    0F                                                         01100000
         DC    AL2(APL2CMDL)                    COMMAND LENGTH          01110000
         DC    AL2(APL2CMDO-APL2CMDN)           OFFSET TO OPTIONS       01120000
*-------------------------------------------------------------*         01130000
*        To customize the APL2 command, modify the following  *         01140000
*        two Assembler statements.                            *         01150000
*                                                             *         01160000
*        The command name defined by APL2CMDN must be a 9-byte*         01170000
*        character constant.  The command options defined by  *         01180000
*        APL2CMDO may be elided, or can be of any length      *         01190000
*        subject to the restrictions of MVS/TSO or VM/CMS.    *         01200000
*        The options can be separated into several DC         *         01210000
*        statements if necessary.                             *         01220000
*-------------------------------------------------------------*         01230000
APL2CMDN DC    CL9'APL2'                        COMMAND                 01240000
APL2CMDO DC    C'QUIET RUN(APL2PI)'             OPTIONS                 01250000
*-------------------------------------------------------------*         01260000
*        End of modifiable statements                         *         01270000
*-------------------------------------------------------------*         01280000
APL2CMDL EQU   *-APL2CMD                                                  01290000
         SPACE 1                                                         01300000
         AP2TAG AP2XAPIC                                                 01310000
         SPACE 1                                                         01320000
         END                                                             01330000
```

DEFAULT VSAPLPR Sample — DEFAULT AP2MSAMP

```
| ************************************************************
| *  APL2 Version 2 Release 2
| *  Licensed Materials - Property of IBM
| *  5688-228, 5688-229 (C) COPYRIGHT IBM CORP. 1984, 1994.
| *  See Copyright Instructions
| ************************************************************
| COLUMN WRAP
| COPY OFF ID * CODE 0
| DISPLAY ON HOLD ON ORIGIN DEFAULT SIZE DEFAULT CODE 0
| LOG SIZE 8140
| PFK 1 IMMEDIATE HELP
| PFK 2 DELAY
| PFK 3 APL DELAY )LOAD
| PFK 4 IMMEDIATE COPY SCREEN
| PFK 5 IMMEDIATE SUPPRESS
| PFK 6 APL DELAY )SAVE
| PFK 7 IMMEDIATE PAGE -1
| PFK 8 IMMEDIATE PAGE +1
| PFK 9 APL DELAY →⎕LC
| PFK 10 IMMEDIATE COLUMN -40
| PFK 11 IMMEDIATE COLUMN +40
| PFK 12 APL DELAY →
| PFK 13 IMMEDIATE HELP
| PFK 14 DELAY
| PFK 15 APL DELAY )LOAD
| PFK 16 IMMEDIATE COPY SCREEN
| PFK 17 IMMEDIATE SUPPRESS
| PFK 18 APL DELAY )SAVE
| PFK 19 IMMEDIATE PAGE -1
| PFK 20 IMMEDIATE PAGE +1
| PFK 21 APL DELAY →⎕LC
| PFK 22 IMMEDIATE COLUMN -40
| PFK 23 IMMEDIATE COLUMN +40
| PFK 24 APL DELAY →
```

LIBTAB APL2 Sample — LIBTAB AP2MSAMP

```
| *
| * LIBTAB for APL2
| *
| * Table is read at session start after AP2EXIT is executed.
| *
| * Card Syntax:
| *   Type Libno Source
| *
| *   Where:  Type  - PRIVATE = filetype APLWSV2
| *                   PUBLIC  = filetype Vnnnnnnn
| *           Libno - n or (n n) where  n ranges from 0 to 999999
| *                   Library numbers are used by AP 121 and APL commands
| *           Source- @ fm = Immediate access
| *                   id cuu write-link-mode = Dynamic LINK and ACCESS
| *                          If optional write-link-mode omitted, W used.
| *
| *   Examples:
| *          PRIVATE (2000 4000) FRED 191 W
| *          PUBLIC   5000       SUE  192 M
| *          PUBLIC   5789       @    B
| *
| * Migration notes:
| *   PROJECT cards are no longer supported as function is part of PUBLIC
| *   PUBLIC  cards differ slightly in format to allow dynamic linking
| *
| ***************************************************************
| *  APL2 Version 2 Release 2
| *  Licensed Materials - Property of IBM
| *  5688-228, 5688-229 (C) COPYRIGHT IBM CORP. 1984, 1994.
| *  See Copyright Instructions
| ***************************************************************
| PUBLIC  (1 99) @ A
| *
| *      Number ranges 1001-1026 will find workspaces on a disk
| *           with the corresponding letter of the alphabet -
| *           1001 = A, 1002 = B, etc.
| *
| PRIVATE 1001 @ A1
| PRIVATE 1002 @ B1
| PRIVATE 1003 @ C1
| PRIVATE 1004 @ D1
| PRIVATE 1005 @ E1
| PRIVATE 1006 @ F1
| PRIVATE 1007 @ G1
| PRIVATE 1008 @ H1
| PRIVATE 1009 @ I1
| PRIVATE 1010 @ J1
| PRIVATE 1011 @ K1
| PRIVATE 1012 @ L1
| PRIVATE 1013 @ M1
| PRIVATE 1014 @ N1
```

```
|  PRIVATE 1015 @ O1
|  PRIVATE 1016 @ P1
|  PRIVATE 1017 @ Q1
|  PRIVATE 1018 @ R1
|  PRIVATE 1019 @ S1
|  PRIVATE 1020 @ T1
|  PRIVATE 1021 @ U1
|  PRIVATE 1022 @ V1
|  PRIVATE 1023 @ W1
|  PRIVATE 1024 @ X1
|  PRIVATE 1025 @ Y1
|  PRIVATE 1026 @ Z1
|  *
|  *      If the library 0 card is found when searching for the
|  *      library equal to the user's account identification («°AI)
|  *      it will be used as the definition for that library.
|  *
|  PRIVATE 0    @ A
```

NAMESYS Macro Sample — NAMESYS AP2MSAMP

```
| ************************************************************        00010000
| *                                                                   00370000
| *   APL2 Version 2 Release 2                                        00380000
| *   Licensed Materials - Property of IBM                            00390000
| *   5688-228, 5688-229 (C) COPYRIGHT IBM CORP. 1984, 1994.          00400000
| *   See Copyright Instructions                                     00410000
| *                                                                   00420000
| ************************************************************        00430000
| *                                                                   00020000
| *   Sample NAMESYS macros for defining the APL2 saved               00030000
| *   segments in DMKSNT ASSEMBLE, as described in the                00040000
| *   "Virtual Machine/System Product: System Programmer's            00050000
| *   Guide" (SC19-6203).  You must supply values appropriate         00060000
| *   on your system for the SYSVOL and SYSSTRT parameters.           00070000
| *                                                                   00080000
| *   Similar DEFSEG commands for VM systems which use DEFSEG         00090000
| *   to define saved segments are:                                   00100000
| *                                                                   00110000
| *      DEFSEG AP2V22V1 600-64F   SR                                 00120000
| *      DEFSEG AP2V22W1 6000-60FF SR                                 00130000
| *      DEFSEG AP2V22E1 600-63F   SR                                 00140000
| *      DEFSEG AP2V22F1 6000-60CF SR                                 00150000
| *      DEFSEG AP2SM22  7000-7FFF SW                                 00160000
| *                                                                   00170000
| *   In this case some of the segments have been defined at          00180000
| *   addresses above 16 megabytes, and the GSVP segment has          00190000
| *   been defined with the maximum size supported, 16 megabytes.    00200000
| *   AP2V22V1 and AP2V22E1, if defined, must continue to reside      00210000
| *   below 16 megabytes.                                             00220000
| *                                                                   00230000
| *   DEFSEG is described in "IBM VM/XA System Product                00240000
| *   Installation and Service" (SC23-0364).                         00250000
| *                                                                   00260000
| *   Some VM systems support OVERRIDE source files as a means        00270000
| *   of defining segments, using the DEFSEG tag.  DEFSEG tag         00280000
| *   parameters match parameters used on NAMESYS macros.             00290000
| *   Substitute values such as shown here with NAMESYS when          00300000
| *   using DEFSEG tags in OVERRIDE files.                           00310000
| *                                                                   00320000
| *   OVERRIDE source files are described in "IBM VM/ESA CP           00330000
| *   Planning and Administration for 370" (SC24-5430).              00340000
| *                                                                   00350000
| ************************************************************        00360000
| *                                                                   00440000
| *        APL2  Shared Code             - Addr = 600000-74FFFF       00450000
| *                                                                   00460000
| AP2V22V1 NAMESYS SYSSIZE=320K,                                     +00470000
|                SYSNAME=AP2V22V1,VSYSADR=IGNORE,VSYSRES=,           +00480000
|                SYSVOL=ssssss,SYSCYL=,SYSSTRT=(ccc,p),              +00490000
|                SYSPGCT=80,SYSPGNM=(1536-1615),                     +00500000
```

```
|               SYSHRSG=(96,97,98,99,100)                                 00510000
|  AP2V22W1 NAMESYS SYSSIZE=1024K,                                       +00520000
|               SYSNAME=AP2V22W1,VSYSADR=IGNORE,VSYSRES=,                +00530000
|               SYSVOL=ssssss,SYSCYL=,SYSSTRT=(ccc,p),                   +00540000
|               SYSPGCT=256,SYSPGNM=(1616-1871),                         +00550000
|               SYSHRSG=(101,102,103,104,105,106,107,108,                +00560000
|               109,110,111,112,113,114,115,116)                          00570000
|  *                                                                      00580000
|  * APL2 Execute-Only Shared Code        - Addr = 600000-70FFFF           00590000
|  *                                                                      00600000
|  AP2V22E1 NAMESYS SYSSIZE=256K,                                        +00610000
|               SYSNAME=AP2V22E1,VSYSADR=IGNORE,VSYSRES=,                +00620000
|               SYSVOL=ssssss,SYSCYL=,SYSSTRT=(ccc,p),                   +00630000
|               SYSPGCT=64,SYSPGNM=(1536-1599),                          +00640000
|               SYSHRSG=(96,97,98,99)                                     00650000
|  AP2V22F1 NAMESYS SYSSIZE=832K,                                        +00660000
|               SYSNAME=AP2V22F1,VSYSADR=IGNORE,VSYSRES=,                +00670000
|               SYSVOL=ssssss,SYSCYL=,SYSSTRT=(ccc,p),                   +00680000
|               SYSPGCT=208,SYSPGNM=(1600-1807),                         +00690000
|               SYSHRSG=(100,101,102,103,104,105,106,107,                +00700000
|               108,109,110,111,112)                                      00710000
|  *                                                                      00720000
|  *        APL2  Global Shared Memory    - Addr = 750000-7FFFFF          00730000
|  *                                                                      00740000
|  *        PROTECT=OFF required to share writable memory                 00750000
|  *        SYSSIZE determined by number of users and size of data        00760000
|  *        SYSNAME value is the only security feature                    00770000
|  *                                                                      00780000
|  AP2SM22  NAMESYS SYSSIZE=704K,PROTECT=OFF,                            +00790000
|               SYSNAME=AP2SM22,VSYSADR=IGNORE,VSYSRES=,                 +00800000
|               SYSVOL=ssssss,SYSCYL=,SYSSTRT=(ccc,p),                   +00810000
|               SYSPGCT=176,SYSPGNM=(1872-2047),                         +00820000
|               SYSHRSG=(117,118,119,120,121,122,123,124,                +00830000
|               125,126,127)                                              00840000
```

NAME3800 Macro Sample — NAME3800 AP2MSAMP

```
| ************************************************************
| *  APL2 Version 2 Release 2
| *  Licensed Materials - Property of IBM
| *  5688-228, 5688-229 (C) COPYRIGHT IBM CORP. 1984, 1994.
| *  See Copyright Instructions
| ************************************************************
| *  Sample NAME3800 macro for defining a named system for                00001000
| *  APL2 fonts in DMKSNT ASSEMBLE, as described in the                   00002000
| *  "Virtual Machine/System Product: Planning Guide and                 00003000
| *  Reference" (SC19-6201).  You must supply values appropriate          00004000
| *  on your system for the SYSVOL and SYSSTRT parameters.                00005000
| ************************************************************          00007000
| *                                                                     00014000
| *        APL2  3800 Font Definition                                    00015000
| *                                                                     00016000
| AP2M3800 NAME3800 CPNAME=AP2F3800      <- name for IMAGELIB, IMAGEMOD  +00017000
|                SYSVOL=ssssss,                                        +00018000
|                SYSSTRT=(ccc,p),                                      +00019000
|                SYSPGCT=100             <- includes room for growth     00020000
```

RDEVICE Macro Sample — RDEVICE AP2MSAMP

```
| ************************************************************
| *  APL2 Version 2 Release 2
| *  Licensed Materials - Property of IBM
| *  5688-228, 5688-229 (C) COPYRIGHT IBM CORP. 1984, 1994.
| *  See Copyright Instructions
| ************************************************************          00006000
| *  Sample RDEVICE macro for defining a 3800 in                        00001000
| *  DMKRIO ASSEMBLE, as described in the "Virtual                      00002000
| *  Machine/System Product: Planning Guide and Reference"             00003000
| *  (SC19-6201).                                                       00004000
| ************************************************************          00012000
| *                                                                     00013000
| *        APL2 3800 Sample RDEVICE                                     00014000
| *                                                                     00015000
| AP2M3800 RDEVICE ADDRESS=cuu,         <- See also RCTLUNIT, RCHANNEL  +00016000
|                FEATURE=4WCGMS,        <- Delete if 2WCGM 3800         +00017000
|                IMAGE=AP2F3800,        <- IMAGELIB at Cold Start       +00018000
|                CHARS=AT10,            <- FONT at Cold Start           +00019000
|                DPMSIZE=1              <- Delayed Purge (1 to 9)        00020000
```

Appendix C. Sample Installation

Sample APL2 Installation

```
*
*  Following is a sample console listing from an uncustomized
*  installation.
*
*  Saved segments have not as yet been defined for APL2, so
*  messages about segments being "UNAVAILABLE" and "UNUSABLE"
*  are normal.
*
*  If the segments are later defined, the APL2 product can be
*  saved in them with the command:
*
*      AP2MAINT APL2SEG
*
*  An AP2LOCAL NAMES file was created before APL2 was
*  installed, and GDDMSEGS and SQLSEGS :NICKS were defined
*  to identify the saved segments used by those products
*  when called from APL2.
*
*  If this is not done the verification exec, V5688228, has no
*  way to detect possible segment overlaps, and does not verify
*  the parts of APL2 that actually use GDDM or SQL/DS.
*  See the description of the AP2LOCAL AP2MSAMP for more
*  information.
*
Ready; T=0.01/0.01 15:44:49
vmfplc2 load * * ( eof 2
 Loading ...
 I5688228 022009   A1
 I5688228 EXEC     A2
 End-of-file or end-of-tape
 I5688228 MEMO     A2
 End-of-file or end-of-tape
Ready; T=0.02/0.05 15:44:58
print i5688228 memo a ( cc
PRT FILE 4610 SENT FROM APL2     PRT WAS 4610 RECS 0216 CPY  001 A NOHOLD NOKEEP
Ready; T=0.01/0.02 15:45:15
i5688228 install
I5688228667  APL2 INSTALL STARTING
I5688228667  APL2 LOAD STARTING
I5688228674  TAPE FILE LOADED, FILE 3 - INSTALLATION, SERVICE, AND SAMPLE FILES
I5688228674  TAPE FILE LOADED, FILE 4 - ONLINE HELP
I5688228674  TAPE FILE LOADED, FILE 5 - RUN-TIME AUXILIARY FILES
I5688228674  TAPE FILE LOADED, FILE 6 - WORKSPACES, WS IMAGES
I5688228674  TAPE FILE SKIPPED, FILE 7 - MACROS, TEXT DECKS
I5688228674  TAPE FILE SKIPPED, FILE 8 - MIGRATION AIDS FOR VS APL APS
I5688228674  TAPE FILE SKIPPED, FILE 9 - 3800 EXECS AND FONTS
I5688228670  APL2 LOAD SUCCESSFUL
I5688228667  APL2 GENERATE STARTING
```


© Copyright IBM Corp. 1984, 1994

```
  I5688228675  APL2 SYSTEM GENERATION FOR:
| AP2MAINT662  LEVEL 220 ( SAVESEGS DISK=A2 GDDM SQL CSVP NOF3800 NOLOCAL )
  DMSCPY721I Copy AP2V220 MODULE A2 to APL2 MODULE A2 (new file)
  DMSCPY721I Copy AP2V220 LOADLIB A2 to APL2 LOADLIB A2 (new file)
| DMSCPY721I Copy AP2W220 LOADLIB A2 to AP2HIGH LOADLIB A2 (new file)
| AP2MAINT662  DEFAULT SYSTEM - APL2 LEVEL 220
  AP2VIDCS610 SERVICE (SEGMENT AP2V22V1) UNAVAILABLE
  AP2MLNK660   SERVICE NSS AP2V22V1 UNUSABLE ( NOT DEFINED )
  AP2VIDCS610 SERVICE (SEGMENT AP2V22W1) UNAVAILABLE
  AP2MLNK660   SERVICE NSS AP2V22W1 UNUSABLE ( NOT DEFINED )
| DMSCPY721I Copy AP2C220 LOADLIB A2 to AP2CSVP LOADLIB A2 (new file)
  AP2VIDCS610 SERVICE (SEGMENT AP2SM22) UNAVAILABLE
  AP2MLNK660   SERVICE NSS AP2SM22 UNUSABLE ( NOT DEFINED )
  I5688228670  APL2 GENERATION SUCCESSFUL
  V5688228667  APL2 VERIFICATION STARTING
  V5688228668  USE THE 'CLEAR' KEY TO CLEAR YOUR DISPLAY SCREEN
  V5688228669  THE 'PA2' KEY MEANS SUPPRESS OUTPUT TO APL2
  V5688228665  APL2 SM(OFF) DE(33) ID(1001) SY(35) WS(512K) RUN(CMSIVP) IN( '''      )OFF''' ')OFF' )
  AP2EXIT665   AP2EXIT ON APL2 00001001 GRAF 00000033

| LICENSED MATERIALS - PROPERTY OF IBM  5688-228, 5688-229
| (C) COPYRIGHT IBM CORP. 1984, 1994.  ALL RIGHTS RESERVED.
| US GOVERNMENT USERS RESTRICTED RIGHTS - USE, DUPLICATION OR
| DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
| * IBM - TRADEMARK OF INTERNATIONAL BUSINESS MACHINES

|      APL2 2.2.00 (English)
|      Version 2 Release 2

| CLEAR WS
| 1
|      APL2 INSTALLATION VERIFICATION PROGRAM
|      5688-228, 5688-229

|      THIS PROGRAM WILL DISPLAY INFORMATION DETERMINED
|      ABOUT THE INSTALLATION OF THIS APL2 SYSTEM.
|      INDENTED MESSAGES ARE INFORMATION ONLY, WHILE MESSAGES
|      PRECEDED BY '>>>' SHOULD BE INVESTIGATED AS POSSIBLE
|      INDICATION OF AN INCORRECT INSTALLATION.

|      THE INSTALLATION PROCESS IS DESCRIBED IN "APL2
|      INSTALLATION AND CUSTOMIZATION UNDER CMS" (SH21-1062).

|      HOST ENVIRONMENT
|      ----------------

|      VM/ESA SYSTEM: CP 2.0.0   CMS 9.0

|      VECTOR FACILITY IN USE

|      APL2 PROGRAM PRODUCT NUMBER 5688-228
|      APL2 SYSTEM LEVEL: 2.2.00

|      APL2 EXECUTING OUTSIDE USER ADDRESS SPACE (SAVED SEGMENT NAME AP2V22V1)
|      AP2V22V1 - 00600000 TO 0064FFFF
|      AP2HIGH EXECUTING OUTSIDE USER ADDRESS SPACE (SAVED SEGMENT NAME AP2V22W1)
|      AP2V22W1 - 03C00000 TO 03CFFFFF
|      AP2CSVP EXECUTING OUTSIDE USER ADDRESS SPACE (SAVED SEGMENT NAME AP2SM22 )
```

```
|      AP2SM22  - 03D00000 TO 03DFFFFF

|      VSAM SAVED SEGMENTS AVAILABLE:
|      CMSVSAM  - 00A30000 TO 00BFFFFF
|      CMSAMS   - 00A00000 TO 00B9FFFF
|      CMSDOS   - 00B00000 TO 00B0FFFF
|      CMSBAM   - 00B10000 TO 00B3FFFF

|      GDDM SAVED SEGMENT(S) LISTED IN NAMES FILE: ADMXSS00 ADMXA230
|      GDDM SAVED SEGMENTS AVAILABLE:
|      ADMXSS00 - 00700000 TO 009BFFFF
|      ADMXA230 - 03900000 TO 03BFFFFF

|      SQL SAVED SEGMENT(S) LISTED IN NAMES FILE: NONE

| >>> SAVED SEGMENT: OVERLAPPING SEGMENTS
|      CMSAMS   - 00A00000 TO 00B9FFFF
|      CMSVSAM  - 00A30000 TO 00BFFFFF
|      CMSDOS   - 00B00000 TO 00B0FFFF
|      CMSBAM   - 00B10000 TO 00B3FFFF

|      SHARED MEMORY: GLOBAL
|      GLOBAL SVP SERVER: AP2SVP

|      ASSOCIATED PROCESSORS:
|      ----------------------

|      VERIFYING REXX PROCESSOR (10 NA)
|      REXX PROCESSOR (10 NA) VERIFIED

|      VERIFYING EXTERNAL FUNCTION PROCESSOR (11 NA)
|      EXTERNAL FUNCTION PROCESSOR (11 NA) VERIFIED

|      VERIFYING FILE PROCESSOR (12 NA)
|      FILE PROCESSOR (12 NA) VERIFIED

|      AUXILIARY PROCESSORS:
|      ---------------------

|      APL2 APS SIGNED ON TO SVP:
|       100 101 102 104 110 111 119 120 121 123 124 126 127 211

|      APL2 APS NOT SIGNED ON:
|        NONE

|      VERIFICATION OF APL2 APS:

|      VERIFYING AP100 HOST COMMAND AUXILIARY PROCESSOR
|      AP100 VERIFIED

|      VERIFYING AP101 ALTERNATE INPUT AUXILIARY PROCESSOR
|      AP101 VERIFIED

|      VERIFYING AP102 STORAGE ACCESS AUXILIARY PROCESSOR
|      AP102 VERIFIED

|      VERIFYING AP104 (ONLY FOR INTERNAL USE BY APL2)
|      AP104 VERIFIED

|      VERIFYING AP110 CMS FILE AUXILIARY PROCESSOR
```

```
|      AP110 VERIFIED

|      VERIFYING AP111 QSAM AUXILIARY PROCESSOR
|      AP111 VERIFIED

|      VERIFYING AP119 TCP/IP AUXILIARY PROCESSOR
|      AP119 VERIFIED

|      VERIFYING AP120 APL2 SESSION MANAGER COMMAND AUXILIARY PROCESSOR
|      AP120 VERIFIED

|      VERIFYING AP121 APL DATA FILE AUXILIARY PROCESSOR
|      AP121 VERIFIED

|      VERIFYING AP123 VSAM AUXILIARY PROCESSOR
|      AP123 VERIFIED

|      VERIFYING AP124 FULL SCREEN MANAGEMENT AUXILIARY PROCESSOR
|      AP124 VERIFIED

|      VERIFYING AP126 GDDM AUXILIARY PROCESSOR
|      AP126 VERIFIED

|      VERIFYING AP127 SQL AUXILIARY PROCESSOR
|      SYSTEM CATALOG ENTRY FOR APL2:
|        AP2V2R2C SQLDBA    2 ¯31936 ¯31934 48384 12/16/93.18:01:44 Y X
|        AP2V2R2R SQLDBA    2 ¯31900 ¯31898 39424 12/16/93.18:01:38 Y X
|      AP127 VERIFIED

|      VERIFYING AP211 APL2 OBJECT FILE AUXILIARY PROCESSOR
|      AP211 VERIFIED

|      APL2 SESSION MANAGER:
|      ---------------------
|      AVAILABLE, IN USE (PSS NOT IN USE)

|      WORKSPACE LIBRARY:
|      ------------------

|      APL2 WORKSPACE  1 DISPLAY  : WS IN PLACE
|      APL2 WORKSPACE  1 EXAMPLES : WS IN PLACE
|      APL2 WORKSPACE  1 MATHFNS  : WS IN PLACE
|      APL2 WORKSPACE  1 MEDIT    : WS IN PLACE
|      APL2 WORKSPACE  1 SUPPLIED : WS IN PLACE
|      APL2 WORKSPACE  1 UTILITY  : WS IN PLACE
|      APL2 WORKSPACE  1 WSINFO   : WS IN PLACE
|      APL2 WORKSPACE  2 APLDATA  : WS IN PLACE
|      APL2 WORKSPACE  2 CHARTX   : WS IN PLACE
|      APL2 WORKSPACE  2 CMS      : WS IN PLACE
|      APL2 WORKSPACE  2 FSC126   : WS IN PLACE
|      APL2 WORKSPACE  2 FSM      : WS IN PLACE
|      APL2 WORKSPACE  2 GRAPHPAK : WS IN PLACE
|      APL2 WORKSPACE  2 GDMX     : WS IN PLACE
|      APL2 WORKSPACE  2 PRINTWS  : WS IN PLACE
|      APL2 WORKSPACE  2 SQL      : WS IN PLACE
|      APL2 WORKSPACE  2 TRANSFER : WS IN PLACE
|      APL2 WORKSPACE  2 VAPLFILE : WS IN PLACE
|      APL2 WORKSPACE  2 VSAMDATA : WS IN PLACE
```

```
|      END OF INSTALLATION VERIFICATION PROGRAM.

  V5688228670  APL2 VERIFICATION SUCCESSFUL
  Ready; T=5.42/9.17 15:48:13
  sp console close
```

Sample Application Environment Installation

```
*
* Following is a sample console listing from an uncustomized installation on a system with
* several GDDM releases but no SQL/DS.
*
* The GDDM saved segments overlap, but this does not cause problems unless
* a program using one release of GDDM calls another program
* that uses another release of GDDM.  When GDDM is called by the
* second program, GDDM overlays the shared segment used by GDDM
* under the first program.  When the second program terminates and
* returns to the first program, unpredictable results may occur.
*
* Because the installation was not customized to remove APL interfaces
* to SQL/DS, the CMSIVP package attempts to verify AP 127.  The messages
* produced below following the "Verifying AP127 ..." message are
* normal in this case.
*
q v stor
STORAGE = 08192K
R; T=0.01/0.01 15:25:27
vmfplc2 load * * ( eof 2
 LOADING.....
 I5688003 022007   A1
 I5688003 EXEC     A2
 END-OF-FILE OR END-OF-TAPE
 I5688003 MEMO     A2
 END-OF-FILE OR END-OF-TAPE
R; T=0.03/0.07 15:27:11
print i5688003 memo ( cc
PRT FILE 3897  FOR AITON    COPY 001   NOHOLD
R; T=0.02/0.05 15:27:47
i5688003 install
I5688003667  APL2AE INSTALL STARTING
I5688003667  APL2AE LOAD STARTING
I5688003674  TAPE FILE LOADED, FILE 3 - INSTALLATION, SERVICE, AND SAMPLE FILES
I5688003674  TAPE FILE LOADED, FILE 4 - ONLINE HELP
I5688003674  TAPE FILE LOADED, FILE 5 - RUN-TIME AUXILIARY FILES
I5688003674  TAPE FILE SKIPPED, FILE 6 - WORKSPACES, WS IMAGES
I5688003674  TAPE FILE SKIPPED, FILE 7 - MACROS, TEXT DECKS
I5688003670  APL2AE LOAD SUCCESSFUL
I5688003667  APL2AE GENERATE STARTING
I5688003675  APL2AE SYSTEM GENERATION FOR:
AP2MAINT662  LEVEL 220 ( SAVESEGS DISK=A2 GDDM SQL CSVP NOF3800 NOLOCAL )
DMSCPY721I COPY 'AP2E220 MODULE A2' TO 'APL2AE MODULE A2' FILE).
DMSCPY721I COPY 'AP2E220 LOADLIB A2' TO 'APL2AE LOADLIB A2' (NEW FILE).
AP2MAINT662  DEFAULT SYSTEM - APL2AE LEVEL 220
SYSTEM SAVED
DMSCPY721I COPY 'AP2C220 LOADLIB A2' TO 'AP2CSVP LOADLIB A2' (NEW FILE).
SYSTEM SAVED
I5688003670  APL2AE GENERATION SUCCESSFUL
V5688003667  APL2 APPLICATION ENVIRONMENT VERIFICATION STARTING
V5688003665  APL2AE SM(OFF) DE(33) ID(1001) SY(35) WS(512K) RUN(CMSIVP)
```

```
| LICENSED MATERIALS - PROPERTY OF IBM  5688-228, 5688-229
| (C) COPYRIGHT IBM CORP. 1984, 1992.  ALL RIGHTS RESERVED.
| US GOVERNMENT USERS RESTRICTED RIGHTS - USE, DUPLICATION OR
| DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
| * IBM - TRADEMARK OF INTERNATIONAL BUSINESS MACHINES

|     APL2 2.2.00 (English)
|     (Application Environment)
|     Version 2 Release 2

| CLEAR WS
| 1
|     APL2 INSTALLATION VERIFICATION PROGRAM
|     5688-228, 5688-229

|     THIS PROGRAM WILL DISPLAY INFORMATION DETERMINED
|     ABOUT THE INSTALLATION OF THIS APL2 SYSTEM.
|     INDENTED MESSAGES ARE INFORMATION ONLY, WHILE MESSAGES
|     PRECEDED BY '>>>' SHOULD BE INVESTIGATED AS POSSIBLE
|     INDICATION OF AN INCORRECT INSTALLATION.

|     THE INSTALLATION PROCESS IS DESCRIBED IN "APL2
|     INSTALLATION AND CUSTOMIZATION UNDER CMS" (SH21-1062).

|     HOST ENVIRONMENT
|     ----------------

|     VM/ESA SYSTEM: CP 2.0.0   CMS 9.0

|     VECTOR FACILITY IN USE

|     APL2 APPLICATION ENVIRONMENT PROGRAM PRODUCT NUMBER 5688-229
|     APL2 SYSTEM LEVEL: 2.2.00

|     APL2 EXECUTING OUTSIDE USER ADDRESS SPACE (SAVED SEGMENT NAME AP2V22E1)
|     AP2V22E1 - 00600000 TO 0064FFFF
|     AP2HIGH EXECUTING OUTSIDE USER ADDRESS SPACE (SAVED SEGMENT NAME AP2V22F1)
|     AP2V22F1 - 03C00000 TO 03CFFFFF
|     AP2CSVP EXECUTING OUTSIDE USER ADDRESS SPACE (SAVED SEGMENT NAME AP2SM22 )
|     AP2SM22  - 03D00000 TO 03DFFFFF

|     VSAM SAVED SEGMENTS AVAILABLE:
|     CMSVSAM  - 00A30000 TO 00BFFFFF
|     CMSAMS   - 00A00000 TO 00B9FFFF
|     CMSDOS   - 00B00000 TO 00B0FFFF
|     CMSBAM   - 00B10000 TO 00B3FFFF

|     GDDM SAVED SEGMENT(S) LISTED IN NAMES FILE: ADMXSS00 ADMXA230
|     GDDM SAVED SEGMENTS AVAILABLE:
|     ADMXSS00 - 00700000 TO 009BFFFF
|     ADMXA230 - 03900000 TO 03BFFFFF

|     SQL SAVED SEGMENT(S) LISTED IN NAMES FILE: NONE

| >>> SAVED SEGMENT: OVERLAPPING SEGMENTS
|     CMSAMS   - 00A00000 TO 00B9FFFF
|     CMSVSAM  - 00A30000 TO 00BFFFFF
|     CMSDOS   - 00B00000 TO 00B0FFFF
|     CMSBAM   - 00B10000 TO 00B3FFFF
```

```
|      SHARED MEMORY: GLOBAL
|      GLOBAL SVP SERVER: AP2SVP

|      ASSOCIATED PROCESSORS:
|      ----------------------

|      VERIFYING REXX PROCESSOR (10 NA)
|      REXX PROCESSOR (10 NA) VERIFIED

|      VERIFYING EXTERNAL FUNCTION PROCESSOR (11 NA)
|      EXTERNAL FUNCTION PROCESSOR (11 NA) VERIFIED

|      VERIFYING FILE PROCESSOR (12 NA)
|      FILE PROCESSOR (12 NA) VERIFIED

|      AUXILIARY PROCESSORS:
|      ---------------------

|      APL2 APS SIGNED ON TO SVP:
|       100 101 102 110 111 119 121 123 124 126 127 211

|      APL2 APS NOT SIGNED ON:
|        NONE

|      VERIFICATION OF APL2 APS:

|      VERIFYING AP100 HOST COMMAND AUXILIARY PROCESSOR
|      AP100 VERIFIED

|      VERIFYING AP101 ALTERNATE INPUT AUXILIARY PROCESSOR
|      AP101 VERIFIED

|      VERIFYING AP102 STORAGE ACCESS AUXILIARY PROCESSOR
|      AP102 VERIFIED

|      VERIFYING AP110 CMS FILE AUXILIARY PROCESSOR
|      AP110 VERIFIED

|      VERIFYING AP111 QSAM AUXILIARY PROCESSOR
|      AP111 VERIFIED

|      VERIFYING AP119 TCP/IP AUXILIARY PROCESSOR
|      AP119 VERIFIED

|      VERIFYING AP121 APL DATA FILE AUXILIARY PROCESSOR
|      AP121 VERIFIED

|      VERIFYING AP123 VSAM AUXILIARY PROCESSOR
|      AP123 VERIFIED

|      VERIFYING AP124 FULL SCREEN MANAGEMENT AUXILIARY PROCESSOR
|      AP124 VERIFIED

|      VERIFYING AP126 GDDM AUXILIARY PROCESSOR
|      SESSION MANAGER NOT IN USE.  GDDM MAY ERASE 327X SCREEN
|      AP126 VERIFIED

|      VERIFYING AP127 SQL AUXILIARY PROCESSOR
|      SYSTEM CATALOG ENTRY FOR APL2:
```

```
|        AP2V2R2C SQLDBA    2 -31936 -31934 48384 12/16/93.18:01:44 Y X
|        AP2V2R2R SQLDBA    2 -31900 -31898 39424 12/16/93.18:01:38 Y X
|      AP127 VERIFIED

|      VERIFYING AP211 APL2 OBJECT FILE AUXILIARY PROCESSOR
|      AP211 VERIFIED


|      END OF INSTALLATION VERIFICATION PROGRAM.

  V5688003670  APL2 APPLICATION ENVIRONMENT VERIFICATION SUCCESSFUL
  R; T=14.08/16.36 15:30:14
```

Appendix D. APL2 System Options

Selected system options and their default values are shown in Figure 9. The invocation options and their default values are shown in Figure 10 on page 131.

System Options

The AP2VOPT macro is used in the AP2VIOPT ASSEMBLE module for APL2, and in the AP2VIOPE ASSEMBLE module for APL2 Application Environment, to define an installation's system options. The AP2VOPT macro and a sample version of both the AP2VIOPT and the AP2VIOPE are supplied on the installation tape. The procedure for changing system options is described in “Changing the APL2 System and Invocation Options” on page 62.

Figure 9 (Page 1 of 2). Selected System Options Specified by the AP2VOPT Macro

Parameter	Used to:	Default
AP2SHNM	Specify the names of the first APL2 saved segment for various load addresses.	in AP2VIOPT: AP2V22V1, AP2V22V2 in AP2VIOPE: AP2V22E1, AP2V22E2
AP2HSNM	Specify the names of the second APL2 saved segment for various load addresses.	in AP2VIOPT: AP2V22W1, AP2V22W2 in AP2VIOPE: AP2V22F1, AP2V22F2
ATASKS	Specify, by entry point name, auxiliary processors to which access is provided. These processors are written to the APL2 AP interface and are included in the APL2 load module.	
CSVPID	Specify the VM user ID of the virtual machine to be used for the global Shared Variable Processor.	AP2SVP
DEFAULT	Specify start and end addresses of default options character string as A-type address constants.[1]	DEFAULTA, DEFAULTZ
EXITCMD	Specify the command(s) to be executed by APL2 at initiation and termination. (Maximum of two eight-character names allowed.)	EXEC AP2EXIT
GDMPRTY	Specify the GDDM filetype for print files.	ADMPRINT
OVERIDE	Specify start and end addresses of override options character string as A-type address constants.[1]	OVERRIDA, OVERRIDZ
QNLT	Specify initial National Language Translation.	ENU

© Copyright IBM Corp. 1984, 1994

Figure 9 (Page 2 of 2). Selected System Options Specified by the AP2VOPT Macro

Parameter	Used to:	Default
QTZDEC	Specify Time Zone fraction TOD correction. Use this only if APL2 is reporting incorrect local time. (Any nonnegative integer is treated as a decimal fraction to be catenated to QTZINT.)	0
QTZINT	Specify Time Zone hour TOD correction. Use this only if APL2 is reporting incorrect local time. If QTZINT together with QTZDEC is within the range -12 to 12, APL2 uses the value for □*TZ*. Otherwise APL2 computes the time zone itself.	-13
RESAPS	Specify, by entry point name, auxiliary processors to which access is provided. These processors are written to the VS APL Version 1 AP interface (both those distributed with APL2 and those written by you) and are included in the APL2 load module. Entry point names of AP 100 and AP 110 (AP2V100 and AP2V110) are required for the correct execution of the)*HOST*,)*IN*,)*PIN*, and)*OUT* commands.	AP2V100, AP2V101, AP2V102, AP2V110, AP2V111, AP2V123, AP2X120 (not in AP2VIOPE), AP2X121, AP2X124, AP2X126
RESEPS	Specify entry point names that are resident in the APL load module. These can include (but are not limited to) auxiliary processors and external processors such as processor 10, 11, and 12. The names are declared by APL2 as CMS nucleus extensions during APL2 initialization. If the names are entry points for APs, then they must also be specified in the APNAMES invocation option when APL2 is invoked. Entry point name AP2X104 is required for the correct execution of the APL2 system commands that reference workspaces in libraries.	AP2VN10, AP2VN11, AP2VN12, AP2XNRSS, AP2X104 (not in AP2VIOPE), AP2X119, AP2X127, AP2X211

Note:

1 Under DEFAULT and OVERIDE the default values address the start and end of a character string, which consists of a series of C-type character constants near the end of the AP2VIOPT ASSEMBLE source. To change what options are specified, edit the C-type character constants rather than the address names.

Invocation Options

The invocation options are explained in detail in “Continuing with the Invocation Options” on page 26.

Figure 10 (Page 1 of 2). Default Invocation Options

Parameter	Used to:	Default
APNAMES	Specify the auxiliary processors not resident with APL2 (identified with AP2VOPT by ATASKS or RESAPS) but with which you plan to share. Auxiliary processor entry point names identified with AP2VOPT by RESEPS can be included here.	in AP2VIOPT: AP2X104, AP2X119(SERVPORT(31415) TCPID(TCPIP)), AP2X127(ISOL(RR)), AP2X211 in AP2VIOPE: AP2X119(SERVPORT(31415) TCPID(TCPIP)), AP2X127(ISOL(RR)), AP2X211
CASE	Specify the translation to be applied to the underbarred alphabet.	1
DATEFORM	Specify date and time representation convention.	ISO
DBCS	Indicate whether you want mixed APL2/DBCS support for I/O during your APL2 session.	TRY
DEBUG	Specify error recovery actions for debugging purposes. (Negative values set flags off.) See *APL2/370 Programming: System Services Reference* for more information about the DEBUG parameter.	0
DSOPEN	Specify a value to be passed to GDDM on a DSOPEN call.	*
EXCLUDE	Specify the auxiliary processors that you do not want to use during your session. The correct execution of the)*HOST*,)*IN*,)*PIN*, and)*OUT* commands and the system commands that reference workspaces in libraries requires that the names of AP 100, AP 104, and AP 110 (AP2V100, AP2X104 and AP2V110) must *not* be specified in this parameter value.	
FREESIZE	Specify a minimum limit on the amount of virtual storage reserved for other applications at APL2 invocation.	in AP2VIOPT: 256K in AP2VIOPE: 64K
HILIGHT	Specify whether input, output, or both is highlighted on the session manager screen.	INPUT

Figure 10 (Page 2 of 2). Default Invocation Options

Parameter	Used to:	Default
ID	Specify an identifier (default library) number to be associated with your APL2 session.	0
INPUT	Specify one or more strings of data to be initially executed by APL2.	
NLT	Specify the national language to be used. This invocation option can override the installation default.	DEFAULT
PROFILE	Specify the name of the session manager profile to be loaded on invocation.	DEFAULT
QUIET	Prevent APL2 session from displaying output until it prompts for input.	
RUN	Specify the name of a niladic function in a namespace as a part of entry to APL2.	
SHRSIZE	Specify the amount of virtual storage to be reserved for shared variables.	40K
SMAPL	Specify whether the session manager is to be used during an APL2 session.	TRY
SVMAX	Specify the maximum number of shared variables that can be shared concurrently.	88
SYSDEBUG	Specify special debugging settings for use by system programmers. (Negative values set flags off.) See *APL2/370 Diagnosis Guide* for more information about the SYSDEBUG parameter.	-255
TERMCODE	Identify that the terminal is not to be used.	(no default)
TRACE	Specify special trace settings for use by system programmers. (Negative values set flags off.)	-65535
WSSIZE	Specify size of active workspace for APL2 session as number of bytes or % of virtual machine size.	25%
XA	Identify storage addressing required or that can be used.	(no default)

Appendix E. APL2 Files and Data Sets

This appendix lists and defines the default files and data sets that are used with the APL2 Licensed Program. In most cases, the default definitions can be modified by the installation or the individual user. Also included in this list are the files and data sets that are defined by the installation or the individual user for using or modifying APL2.

Files

File name	File type	Description
file name	ADMPRINT[1]	The default file type for print files from the the GDDM Environments Defaults Module for VM.
file name	ADMSYMBL	Source of the APL2 program symbol set used by GDDM.
file name	APLTF	The default file type for transfer files written using the)*IN* and)*OUT* system commands.
APLIBTAB	APLIBTAB	The library table used by the)*MCOPY* system command.
wkspcname	APLWSV2[1]	The default file type for APL2 private library workspaces.
DUMP*nnnn*	APLWSV2	The name of a workspace dump.
language	APL2LANG	National Language translation for messages and command keywords selected by □*NLT*.
language	APL2HELP	National Language translation for help text entries selected by □*NLT*.
AP2TCPIP	APL2PROF	A cross-system shared variable profile.
file name	APL2UT1	The temporary file type assigned to workspace files during)*SAVE* system command processing.
n	AP2EDPGM	File used when editing a function or operator.
n	AP2EDCHR	File used when editing a simple character variable.
n	AP2EDEVL	File used when editing any other variable (for example, a vector of numbers, or a combination of numbers and characters).
SINKWS	AP2UT2	Used by the)*COPY*,)*PCOPY*, and)*MCOPY* system commands.
LIBTAB	APL2	Used to associate a library number to a virtual disk.
AP2EXIT[1]	EXEC	Used during APL2 invocation and termination.
@LOG1 @LOG2	*file type*	Used to store the session manager log. The file name switches from @LOG1 to @LOG2 when you change the session log size. The file type and file mode depend on LIBTAB APL2.
wkspcname	**V***nnnnnnn*	The default file type for APL2 public library workspaces.
file name	VMAPL*F	The default file type for AP 110 files.
file name	VSAPLFL[1] VSAPL@L[1]	The default file type for private-library APL data files; also used for session manager log files and copy work files.
DEFAULT	VSAPLPR[1]	The default APL2 session manager profile.
file name	VSAPLPR[1]	The default file type for the APL2 session manager profile.

© Copyright IBM Corp. 1984, 1994

File name	File type	Description
wkspcname	VSAPLWS[1]	The default file type for VS APL private library workspaces that are used as input to the)*MCOPY* system command.
wkspcname	**W***nnnnnnn*	The default file type for VS APL project and public library workspaces that are used as input to the)*MCOPY* system command.

CMS FILEDEF (DD) Names

Name	Description
$APL2	Used to load nonshared copy of APL2.
APLDUMP	Used to request formatted dumps.
APLIN	Contains input statements acceptable to APL2.
APLPRINT	Contains a log of the APL session. .
AP2LOAD	The DDNAME used to define auxiliary processor load modules.

[1] Name can be modified by the installation.

Appendix F. AP2M220 NAMES File

```
| :rem.*-------------------------------------------------------------*
| *
| * 5688-228, 5688-229  APL2 Installation and Maintenance Definition
| *
| * Copyright: Licensed Materials - Property of IBM
| *            5668-899, 5688-003
| *            (C) COPYRIGHT IBM CORP. 1984, 1994.
| *            See Copyright Instructions.
| *
| * Status:    APL2 Version 2 Release 2
| *
| *-------------------------------------------------------------*

| :rem.*-------------------------------------------------------------*
| *
| *  Each EXEC first reads :NICK.ID to establish the
| *       system level and the name of the installation
| *       look-aside file (AP2LOCAL) which contains
| *       all customization information.
| *
| *  Thus :NICK thereafter is looked for first in :LocalFN
| *                                     then  in this file
| *       This permits substitution of installation data for
| *       anything in this file.
| *
| *  The installation/maintenance process
| *      involves having an AP2M___ EXEC follow the tree
| *      specified by some list of :NICK tags.
| *
| *      Each :NICK will be searched for the first tag (in order) of:
| *
| *             :cmd    - A Command to be executed immediately
| *                       The ';' may be used to delimit clauses.
| *                       If a non-zero return code occurs,
| *                       the active :NICK is abandoned.
| *                       NOTE: This tag may be any EXEC2
| *                             statement executable with &READ 1
| *
| *             :msg    - Arbitrary message to type
| *                       The ';' may be used to delimit lines.
| *
| *             :list   - List of additional :NICKS to process
| *
| *                       Because NAMEFIND has limits, within a :NICK, on
| *                       the number of tags and the length of a tag
| *                       value, lists are frequently used in this file.
| *                       As an informal convention, lists which
| *                       begin with a lowercase letter and hyphen
| *                       (e.g. 'l-', 'c-') are not intended to be
| *                       triggered by a terminal entry, but are really
```

© Copyright IBM Corp. 1984, 1994

```
|  *                      sub-lists to overcome NAMEFIND restrictions.
|  *
|  *--------------------------------------------------------------------*

|  :Nick.ID           :LEVEL.220            :rem. Sets &-LV
|                     :LocalFN.AP2LOCAL

|  :rem.*-------------------------------------------------------------*
|  *
|  *  AP2MAINT - System Installation/Maintenance tags
|  *
|  *--------------------------------------------------------------------*

|  :Nick.APL2-PID     :rem. System as distributed by IBM
|         :cmd.&CALL -<COPY> AP2V&-LV MODULE  * APL2 = &-FM OLDD TYPE ;
|              &CALL -<COPY> AP2V&-LV LOADLIB * APL2 = &-FM OLDD TYPE ;
|              &CALL -<COPY> AP2W&-LV LOADLIB * AP2HIGH = &-FM OLDD TYPE
|                     :msg.DEFAULT SYSTEM - APL2 LEVEL &-LV

|  :Nick.APL2AE-PID   :rem. System as distributed by IBM
|         :cmd.&CALL -<COPY> AP2E&-LV MODULE  * APL2AE = &-FM OLDD TYPE ;
|              &CALL -<COPY> AP2E&-LV LOADLIB * APL2AE = &-FM OLDD TYPE ;
|              &CALL -<COPY> AP2F&-LV LOADLIB * AP2HIAE = &-FM OLDD TYPE
|                     :msg.DEFAULT SYSTEM - APL2AE LEVEL &-LV

|  :Nick.APL2-BASE    :rem. LKED for Local Options, GDDM, SQL
|             :list. m-APL2 l-APL2-BASE l-HIGH-BASE

|  :Nick.APL2AE-BASE  :rem. LKED for Local Options, GDDM, SQL
|             :list. m-APL2AE l-APAE-BASE l-HIAE-BASE

|  :Nick.l-APL2-BASE  :rem. l-APL2-BASE is a faster LKED than l-APL2
|                     :cmd.EXEC AP2MLNK APL2     LOADLIB  APL2 AP2V&-LV
|                     VIAAA    Local-TEXT t-&-GDDM t-&-GDMXBASE (&-Sys
|                     ( RENT LIST MAP NOTERM SIZE 384K 128K AMODE 31 RMODE 24

|  :Nick.l-APAE-BASE  :rem. l-APAE-BASE is a faster LKED than l-APL2AE
|                     :cmd.EXEC AP2MLNK APL2AE LOADLIB  APL2 AP2E&-LV
|                     VIAAA    Local-AETEXT t-&-GDDM t-&-GDMXBASE (&-Sys
|                     ( RENT LIST MAP NOTERM SIZE 384K 128K AMODE 31 RMODE 24

|  :Nick.l-HIGH-BASE  :rem. l-HIGH-BASE is a faster LKED than l-HIGH
|                     :cmd.EXEC AP2MLNK AP2HIGH LOADLIB AP2HIGH AP2W&-LV
|                     VIAAA Local-HIGHTEXT t-&-QMF t-&-APL2SMBASE
|                     t-&-AP127BASE t-&-SQL (&-Sys

|  :Nick.l-HIAE-BASE  :rem. l-HIAE-BASE is a faster LKED than l-HIAE
|                     :cmd.EXEC AP2MLNK AP2HIAE LOADLIB AP2HIGH AP2F&-LV
|                     VIAAA Local-HIAETEXT t-&-QMF
|                     t-&-AP127BASE t-&-SQL (&-Sys

|  :Nick.APL2-PTF     :rem. Full LKED for maintenance
|                     :msg.PATIENCE PLEASE. APL2-PTF TAKES TIME TO RUN
|                     :list. CLEAN m-APL2 l-APL2 l-HIGH
```

```
|  :Nick.APL2AE-PTF   :rem. Full LKED for maintenance
|                     :msg.PATIENCE PLEASE. APL2AE-PTF TAKES TIME TO RUN
|                     :list. CLEANAE m-APL2AE l-APL2AE l-HIGHAE

|  :Nick.CSVP-PID     :rem. Global SVP as distributed by IBM
|                     :cmd.&CALL -<COPY> AP2C&-LV LOADLIB * AP2CSVP = &-FM
|                               OLDD REP TYPE
|                     :list. GSVPSEG

|  :Nick.CSVP-PTF     :rem.Apply maintenance to the global SVP
|                     :list. cleaSVPL RC=0 mod-CSVP RC28=0 l-CSVP GSVPSEG

|  :Nick.UTIL-PTF     :rem.Apply maintenance to utility modules
|                     :list. MOD-PTF LOAD-PTF LANG-PTF HELP-PTF FONT-PTF

|  :Nick.MOD-PTF      :list. mod-XLOAD RC28=0
|                            mod-VUTIL RC28=0 nxDrop
|                            mod-VMCF  RC28=0
|                            mod-APL2PI RC28=0

|  :Nick.LOAD-PTF     :list. load-VNRP1 RC28=0
|                            load-VNRP2 RC28=0
|                            load-VNRP3 RC28=0
|                            load-VNRP4 RC28=0
|                            load-VNRP5 RC28=0
|                            load-VNRPS RC28=0
|                            load-VNRPI RC28=0
|                            load-VNRPR RC28=0

|  :Nick.LANG-PTF     :list.lang-LDAN lang-LDEU lang-LENU lang-LESP
|                           lang-LFIN lang-LFRA lang-LFRC lang-LGRE
|                           lang-LHEB lang-LITA lang-LJAP lang-LKAT
|                           lang-LMAG lang-LNOR lang-LPTG lang-LSVE

|  :Nick.HELP-PTF     :list.help-HDEU help-HENP help-HENU help-HJAP

|  :Nick.FONT-PTF     :list.font-ptf1 font-ptf2 font-ptfp

|  :Nick.font-ptf1    :list.fp1-CA fp1-CB fp1-CD fp1-CF fp1-CH fp1-CJ
|                           fp1-CN fp1-C0 fp1-C6 fp1-C7 fp1-C8 fp1-C9

|  :Nick.font-ptf2    :list.fp2-CA fp2-CB fp2-CD fp2-CF fp2-CH fp2-CJ
|                           fp2-CN fp2-C0 fp2-C6 fp2-C7 fp2-C8 fp2-C9
|                           fp2-XA fp2-XB fp2-XD fp2-XF fp2-XH fp2-XJ
|                           fp2-XN fp2-X0 fp2-X6 fp2-X7 fp2-X8 fp2-X9
|                           fp2-T3
|  :Nick.font-ptfp    :list.fpp-CA fpp-CB fpp-CD fpp-CF fpp-CH fpp-CJ
|                           fpp-CN fpp-C0 fpp-C6 fpp-C7 fpp-C8 fpp-C9
|                           fpp-XA fpp-XB fpp-XD fpp-XF fpp-XH fpp-XJ
|                           fpp-XN fpp-X0 fpp-X6 fpp-X7 fpp-X8 fpp-X9
|                           fpp-T3

|  :Nick.nxDROP       :cmd. NUCXDROP AP2VUTIL
```

```
| :Nick.APL2SEG      :rem. APL2 Saved Segments
|                    :list.Clean-SEG1 d-SEG1 Clean-HIGH1 d-HIGH1

| :Nick.APL2AESEG    :rem. APL2AE Saved Segments
|                    :list.Clean-AESEG1 d-AESEG1 Clean-AEHIGH1 d-AEHIGH1

| :Nick.GSVPSEG      :rem. APL2 Global Shared Variable Processor Segment
|                    :list.Clean-SMSEG d-SMSEG

| :Nick.GDDMSEGS     :rem. GDDM Segment Names for CMSIVP
|                    :list.

| :Nick.SQLSEGS      :rem. SQL/DS Segment Names for CMSIVP
|                    :list.

| :Nick.GEN3800      :rem. Install 3800 support
|                    :cmd.EXEC AP2M3800 GENIMAGE AP2G3800 (&-Sys ;
|                         EXEC AP2M3800 IMAGELIB AP2F3800 (&-Sys
|                    :msg.*** 3800 SUPPORT INSTALLLED

| :Nick.GEN3803      :rem. Install 3803 support
|                    :cmd.EXEC AP2M3800 GENIMAGE AP2G3803 (&-Sys ;
|                         EXEC AP2M3800 IMAGELIB AP2F3803 (&-Sys
|                    :msg.*** 3803 SUPPORT INSTALLLED

| :Nick.EPILOG-AP2MAINT   :rem.* Place holder for local use
| :Nick.PROLOG-AP2MAINT   :rem.* Place holder for local use (LINK, etc.)
| :Nick.EPILOG-5688228D   :rem.* Place holder for local use
| :Nick.PROLOG-5688228D   :rem.* Place holder for local use (LINK, etc.)
| :Nick.EPILOG-5688229D   :rem.* Place holder for local use
| :Nick.PROLOG-5688229D   :rem.* Place holder for local use (LINK, etc.)

| :rem.*-------------------------------------------------------------*
| *
| *  AP2MAINT - Service Features and Defaults
| *
| *-------------------------------------------------------------*

| :Nick.CLEAN          :list. Clean-module Clean-load Clean-HIGH RC=0
| :Nick.Clean-module   :cmd.ERASE APL2 MODULE  &-FM  (TYPE
| :Nick.Clean-load     :cmd.ERASE APL2 LOADLIB &-FM  (TYPE
| :Nick.Clean-HIGH     :cmd.ERASE AP2HIGH LOADLIB &-FM  (TYPE
| :Nick.CLEANAE        :list. Clean-AEmod Clean-AEload Clean-HIAE RC=0
| :Nick.Clean-AEmod    :cmd.ERASE APL2AE MODULE  &-FM  (TYPE
| :Nick.Clean-AEload   :cmd.ERASE APL2AE LOADLIB &-FM  (TYPE
| :Nick.Clean-HIAE     :cmd.ERASE AP2HIAE LOADLIB &-FM  (TYPE
| :Nick.cleaSVPL       :cmd.ERASE AP2CSVP  LOADLIB &-FM  (TYPE
| :Nick.Clean-SEG1     :rem.* Place holder for local use (PURGE, DEFSEG)
| :Nick.Clean-SEG2     :rem.* Place holder for local use (PURGE, DEFSEG)
| :Nick.Clean-HIGH1    :rem.* Place holder for local use (PURGE, DEFSEG)
| :Nick.Clean-HIGH2    :rem.* Place holder for local use (PURGE, DEFSEG)
| :Nick.Clean-AESEG1   :rem.* Place holder for local use (PURGE, DEFSEG)
| :Nick.Clean-AESEG2   :rem.* Place holder for local use (PURGE, DEFSEG)
```

```
|  :Nick.Clean-AEHIGH1  :rem.* Place holder for local use (PURGE, DEFSEG)
|  :Nick.Clean-AEHIGH2  :rem.* Place holder for local use (PURGE, DEFSEG)
|  :Nick.Clean-SMSEG    :rem.* Place holder for local use (PURGE, DEFSEG)

|  :Nick.DEBUG        :cmd.&+DEBUG = 1
|  :Nick.TRACE        :cmd.&+TRACE = 1
|  :Nick.RC=0         :cmd.&+EXITRC = 0
|  :Nick.RC28=0       :cmd.&IF &+EXITRC = 28 &+EXITRC = 0

|  :Nick.Quiet        :rem.* Placeholder
|  :Nick.msg-OPTS     :rem.* Tell I5688228 what variables we are using.
|                     :msg.LEVEL &-LV ( &-SEGS
|                          DISK=&-FM &-GDDM &-SQL
|                                    &-CSVP &-F3800 &-LOCAL )

|  :rem.*-------------------------------------------------------------------*
|  *
|  *  Variables
|  *              In general, AP2M___ EXECs use variables starting with
|  *              a hyphen (e.g. &-LV) to substitute in command strings
|  *              built and executed from the data in this NAMES file,
|  *              and names with a plus (e.g. &+TRACE) for debugging.
|  *
|  *-------------------------------------------------------------------*

|  :Nick.IBM-Defaults :list. FM=A GDDM SQL CSVP NOLOCAL
|                                NOF3800 SEGS-cmd

|  :Nick.FM=A         :cmd.&-FM    = A2

|  :Nick.SEGS-cmd     :cmd.&-SEGS  = SAVESEGS

|  :Nick.GDDM         :list.GDDM1 GDDM2 GDDM3 GDDM4 GDDM5
|  :Nick.GDDM1        :cmd.&-GDDM = GDDM
|  :Nick.GDDM2        :cmd.&-APL2SM = APL2SM
|  :Nick.GDDM3        :cmd.&-GDMX = GDMX
|  :Nick.GDDM4        :cmd.&-APL2SMBASE = NULL
|  :Nick.GDDM5        :cmd.&-GDMXBASE = NULL
|  :Nick.t-NULL       :rem.Do nothing
|  :Nick.SQL          :list.SQL1 SQL2 SQL3 SQL4
|  :Nick.SQL1         :cmd.&-QMF   = QMF
|  :Nick.SQL2         :cmd.&-SQL   = SQL
|  :Nick.SQL3         :cmd.&-AP127 = AP127
|  :Nick.SQL4         :cmd.&-AP127BASE = NULL
|  :Nick.CSVP         :cmd.&-CSVP  = CSVP
|  :Nick.F3800        :cmd.&-F3800 = F3800
|  :Nick.LOCAL        :cmd.&-LOCAL = LOCAL

|  :Nick.NOGDDM       :list.NOGDDM1 NOGDDM2 NOGDDM3 NOGDDM4 NOGDDM5
|  :Nick.NOGDDM1      :cmd.&-GDDM = NOGDDM
|  :Nick.NOGDDM2      :cmd.&-APL2SM = NOAPL2SM
|  :Nick.NOGDDM3      :cmd.&-GDMX = NOGDMX
|  :Nick.NOGDDM4      :cmd.&-APL2SMBASE = NOAPL2SM
|  :Nick.NOGDDM5      :cmd.&-GDMXBASE = NOGDMX
```

```
|  :Nick.NOSQL        :list.NOSQL1 NOSQL2 NOSQL3 NOSQL4
|  :Nick.NOSQL1       :cmd.&-QMF   = NOQMF
|  :Nick.NOSQL2       :cmd.&-SQL   = NOSQL
|  :Nick.NOSQL3       :cmd.&-AP127 = NOAP127
|  :Nick.NOSQL4       :cmd.&-AP127BASE = NOAP127
|  :Nick.NOCSVP       :cmd.&-CSVP  = NOCSVP
|  :Nick.NOF3800      :cmd.&-F3800 = NOF3800
|  :Nick.NOLOCAL      :cmd.&-LOCAL = NOLOCAL

|  :Nick.SETSYS       :cmd.&-Sys = &STRING OF &-FM &-LV &+TRACE
|  :Nick.STACK        :cmd.&+DROPBUF = 0 ;
|                          &STACK &-SEGS &-FM &-LV &-GDDM &-SQL
|                          &-CSVP &-F3800 &-LOCAL

|  :rem.*-------------------------------------------------------------*
|  *
|  *  MAINTENANCE TAGS
|  *
|  *-------------------------------------------------------------*

|  :rem.*-------------------------------------------------------------*
|  *
|  *  Informal Convention for Names
|  *
|  *        m- = GENMOD  (AP2MLNK parts are operand  1 MODULE-  )
|  *        l- = LKED    (AP2MLNK parts are operand  1 LOADLIB- )
|  *        d- = SAVESYS (AP2MLNK loads operands 3,4 -member,LOADLIB)
|  *        mod- = do m- GENMOD if updated TEXT decks exist
|  *
|  *-------------------------------------------------------------*

|  :Nick.m-XLOAD    :cmd.EXEC AP2MLNK AP2XLOAD MODULE (&-Sys
|                   (ORIGIN TRANS NOMAP NOAUTO RLDSAVE ( SYSTEM

|  :Nick.m-VUTIL    :cmd.EXEC AP2MLNK AP2VUTIL MODULE (&-Sys
|                   (ORIGIN TRANS NOMAP NOAUTO RLDSAVE

|  :Nick.m-APL2PI   :cmd.EXEC AP2MLNK APL2PI   MODULE VAPI Local-XAPIC
|                     (&-Sys (CLEAR RLDSAVE NOMAP NOAUTO

|  :Nick.m-APL2     :cmd.EXEC AP2MLNK APL2     MODULE  (&-Sys
|  :Nick.m-APL2AE   :cmd.EXEC AP2MLNK APL2AE   MODULE  (&-Sys

|  :Nick.l-APL2     :cmd.EXEC AP2MLNK APL2     LOADLIB  APL2 AP2V&-LV
|                                     VIAAA    Local-TEXT  c-BASE
|                                     t-&-GDDM t-&-GDMX (&-Sys
|                   ( RENT LIST MAP NOTERM SIZE 384K 128K AMODE 31 RMODE 24

|  :Nick.l-APL2AE   :cmd.EXEC AP2MLNK APL2AE   LOADLIB  APL2 AP2E&-LV
|                                     VIAAA    Local-AETEXT  c-AEBASE
|                                     t-&-GDDM t-&-GDMX (&-Sys
|                   ( RENT LIST MAP NOTERM SIZE 384K 128K AMODE 31 RMODE 24

|  :Nick.l-HIGH     :cmd.EXEC AP2MLNK AP2HIGH LOADLIB AP2HIGH AP2W&-LV
```

```
|                                  VIAAA VIOPT Local-HIGHTEXT c-Language
|                                  c-HIX t-&-QMF t-&-APL2SM c-SVP c-HIAP
|                                  t-&-AP127 t-&-SQL VIZZZ (&-Sys

|  :Nick.l-HIGHAE   :cmd.EXEC AP2MLNK AP2HIAE LOADLIB AP2HIGH AP2F&-LV
|                                  VIAAA VIOPE Local-HIAETEXT c-AElang
|                                  c-HIX t-&-QMF c-SVP c-HIAPAE
|                                  t-&-AP127 t-&-SQL VIZZZ (&-Sys

|  :Nick.m-V220     :cmd.EXEC AP2MLNK AP2V220  MODULE  (&-Sys
|  :Nick.m-E220     :cmd.EXEC AP2MLNK AP2E220  MODULE  (&-Sys
|  :Nick.l-V220     :cmd.EXEC AP2MLNK AP2V220  LOADLIB  APL2  (&-Sys
|                   ( RENT LIST MAP NOTERM SIZE 384K 128K AMODE 31 RMODE 24
|  :Nick.l-E220     :cmd.EXEC AP2MLNK AP2E220  LOADLIB  APL2  (&-Sys
|                   ( RENT LIST MAP NOTERM SIZE 384K 128K AMODE 31 RMODE 24
|  :Nick.l-W220     :cmd.EXEC AP2MLNK AP2W220  LOADLIB  AP2HIGH (&-Sys
|  :Nick.l-F220     :cmd.EXEC AP2MLNK AP2F220  LOADLIB  AP2HIGH (&-Sys
|  :Nick.l-C220     :cmd.EXEC AP2MLNK AP2C220  LOADLIB  CSVP  (&-Sys

|  :Nick.d-SEG1     :cmd.EXEC AP2MLNK AP2V22V1 &-SEGS   APL2 APL2 (&-Sys
|  :Nick.d-AESEG1   :cmd.EXEC AP2MLNK AP2V22E1 &-SEGS   APL2 APL2AE (&-Sys
|  :Nick.d-SEG2     :cmd.EXEC AP2MLNK AP2V22V2 &-SEGS   APL2 APL2 (&-Sys
|  :Nick.d-AESEG2   :cmd.EXEC AP2MLNK AP2V22E2 &-SEGS   APL2 APL2AE (&-Sys
|  :Nick.d-HIGH1    :cmd.EXEC AP2MLNK AP2V22W1 &-SEGS AP2HIGH AP2HIGH (&-Sys
|  :Nick.d-AEHIGH1  :cmd.EXEC AP2MLNK AP2V22F1 &-SEGS AP2HIGH AP2HIAE (&-Sys
|  :Nick.d-HIGH2    :cmd.EXEC AP2MLNK AP2V22W2 &-SEGS AP2HIGH AP2HIGH (&-Sys
|  :Nick.d-AEHIGH2  :cmd.EXEC AP2MLNK AP2V22F2 &-SEGS AP2HIGH AP2HIAE (&-Sys

|  :Nick.m-CSVP     :cmd.EXEC AP2MLNK AP2CSVP  MODULE  (&-Sys ( ( SYSTEM
|  :Nick.m-VMCF     :cmd.EXEC AP2MLNK AP2VMCF  MODULE  (&-Sys
|  :Nick.l-CSVP     :cmd.EXEC AP2MLNK AP2CSVP  LOADLIB CSVP AP2C&-LV (&-Sys
|  :Nick.d-SMSEG    :cmd.EXEC AP2MLNK AP2SM22  &-SEGS  CSVP AP2CSVP  (&-Sys

|  :Nick.l-VNRP1    :cmd.EXEC AP2MLNK AP2VNRP1 LOADLIB  AP2PDDT  (&-Sys
|  :Nick.l-VNRP2    :cmd.EXEC AP2MLNK AP2VNRP2 LOADLIB  AP2PPKG  (&-Sys
|  :Nick.l-VNRP3    :cmd.EXEC AP2MLNK AP2VNRP3 LOADLIB  AP2PDFMT (&-Sys
|  :Nick.l-VNRP4    :cmd.EXEC AP2MLNK AP2VNRP4 LOADLIB  AP2PIVP  (&-Sys
|  :Nick.l-VNRP5    :cmd.EXEC AP2MLNK AP2VNRP5 LOADLIB  AP2PAPIW (&-Sys
|  :Nick.l-VNRPS    :cmd.EXEC AP2MLNK AP2VNRPS LOADLIB  AP2PPSRV (&-Sys
|  :Nick.l-VNRPI   :cmd.EXEC AP2MLNK AP2VNRPI LOADLIB  AP2PIDI  (&-Sys
|  :Nick.l-VNRPR   :cmd.EXEC AP2MLNK AP2VNRPR LOADLIB  AP2PRAPL (&-Sys

|  :Nick.load-VNRP1   :rem.Regen the AP2VNRP1 loadlib if necessary
|                     :cmd.STATE AP2PDDT  TEXT
|                     :list. cleVNRP1 RC=0 l-VNRP1
|  :Nick.cleVNRP1     :rem.Erase the old loadlib if it is on the target disk
|                     :cmd.STATE AP2VNRP1 LOADLIB &-FM ;
|                          ERASE AP2VNRP1 LOADLIB &-FM

|  :Nick.load-VNRP2   :rem.Regen the AP2VNRP2 loadlib if necessary
|                     :cmd.STATE AP2PPKG  TEXT
|                     :list. cleVNRP2 RC=0 l-VNRP2
|  :Nick.cleVNRP2     :rem.Erase the old loadlib if it is on the target disk
|                     :cmd.STATE AP2VNRP2 LOADLIB &-FM ;
```

```
|                          ERASE AP2VNRP2 LOADLIB &-FM

|  :Nick.load-VNRP3   :rem.Regen the AP2VNRP3 loadlib if necessary
|                     :cmd.STATE AP2PDFMT TEXT
|                     :list. cleVNRP3 RC=0 l-VNRP3
|  :Nick.cleVNRP3     :rem.Erase the old loadlib if it is on the target disk
|                     :cmd.STATE AP2VNRP3 LOADLIB &-FM ;
|                          ERASE AP2VNRP3 LOADLIB &-FM

|  :Nick.load-VNRP4   :rem.Regen the AP2VNRP4 loadlib if necessary
|                     :cmd.STATE AP2PIVP  TEXT
|                     :list. cleVNRP4 RC=0 l-VNRP4
|  :Nick.cleVNRP4     :rem.Erase the old loadlib if it is on the target disk
|                     :cmd.STATE AP2VNRP4 LOADLIB &-FM ;
|                          ERASE AP2VNRP4 LOADLIB &-FM

|  :Nick.load-VNRP5   :rem.Regen the AP2VNRP5 loadlib if necessary
|                     :cmd.STATE AP2PAPIW TEXT
|                     :list. cleVNRP5 RC=0 l-VNRP5
|  :Nick.cleVNRP5     :rem.Erase the old loadlib if it is on the target disk
|                     :cmd.STATE AP2VNRP5 LOADLIB &-FM ;
|                          ERASE AP2VNRP5 LOADLIB &-FM

|  :Nick.load-VNRPS   :rem.Regen the AP2VNRPS loadlib if necessary
|                     :cmd.STATE AP2PPSRV TEXT
|                     :list. cleVNRPS RC=0 l-VNRPS
|  :Nick.cleVNRPS     :rem.Erase the old loadlib if it is on the target disk
|                     :cmd.STATE AP2VNRPS LOADLIB &-FM ;
|                          ERASE AP2VNRPS LOADLIB &-FM

|  :Nick.load-VNRPI  :rem.Regen the AP2VNRPI loadlib if necessary
|                     :cmd.STATE AP2PIDI TEXT
|                     :list. cleVNRPI RC=0 l-VNRPI
|  :Nick.cleVNRPI     :rem.Erase the old loadlib if it is on the target disk
|                     :cmd.STATE AP2VNRPI LOADLIB &-FM ;
|                          ERASE AP2VNRPI LOADLIB &-FM

|  :Nick.load-VNRPR  :rem.Regen the AP2VNRPR loadlib if necessary
|                     :cmd.STATE AP2PRAPL TEXT
|                     :list. cleVNRPR RC=0 l-VNRPR
|  :Nick.cleVNRPR     :rem.Erase the old loadlib if it is on the target disk
|                     :cmd.STATE AP2VNRPR LOADLIB &-FM ;
|                          ERASE AP2VNRPR LOADLIB &-FM

|  :Nick.mod-CSVP     :rem.Regen the AP2CSVP module if necessary
|                     :cmd.STATE AP2VCSVS TEXT
|                     :list. cleaCSVP RC=0 m-CSVP
|  :Nick.cleaCSVP     :rem.Erase the old module if it is on the target disk
|                     :cmd.STATE AP2CSVP MODULE &-FM ;
|                          ERASE AP2CSVP MODULE &-FM

|  :Nick.mod-APL2PI   :rem.Regen the APL2PI module if necessary
|                     :cmd.STATE AP2VAPI  TEXT
```

```
|                     :list. cleaAPI RC=0 m-APL2PI
|  :Nick.cleaAPI      :rem.Erase the old module if it is on the target disk
|                     :cmd.STATE APL2PI MODULE &-FM ;
|                          ERASE APL2PI MODULE &-FM

|  :Nick.mod-XLOAD    :rem.Regen the AP2XLOAD module if necessary
|                     :cmd.STATE AP2VMRLD TEXT
|                     :list. cleaXLOA RC=0 m-XLOAD
|  :Nick.cleaXLOA     :rem.Erase the old module if it is on the target disk
|                     :cmd.STATE AP2XLOAD MODULE &-FM ;
|                          ERASE AP2XLOAD MODULE &-FM

|  :Nick.mod-VMCF     :rem.Regen the AP2VMCF module if necessary
|                     :cmd.STATE AP2VMEXT TEXT
|                     :list. cleaVMCF RC=0 m-VMCF
|  :Nick.cleaVMCF     :rem.Erase the old module if it is on the target disk
|                     :cmd.STATE AP2VMCF MODULE &-FM ;
|                          ERASE AP2VMCF MODULE &-FM

|  :Nick.mod-VUTIL    :rem.Regen the AP2VUTIL module if necessary
|                     :cmd.AP2XLOAD AP2VUTIL AP2VUTIL ;
|                          STATE AP2VIDCS TEXT
|                     :list. cleaVUTI RC=0 m-VUTIL
|  :Nick.cleaVUTI     :rem.Erase the old module if it is on the target disk
|                     :cmd.STATE AP2VUTIL MODULE &-FM ;
|                          ERASE AP2VUTIL MODULE &-FM

|  :Nick.lang-LDAN    :cmd.EXEC AP2MREN AP2LDAN APL2LANG &-FM DANSK ;
|  :Nick.lang-LDEU    :cmd.EXEC AP2MREN AP2LDEU APL2LANG &-FM DEUTSCH ;
|  :Nick.lang-LENU    :cmd.EXEC AP2MREN AP2LENU APL2LANG &-FM DEFAULT ;
|  :Nick.lang-LESP    :cmd.EXEC AP2MREN AP2LESP APL2LANG &-FM ESPANOL ;
|  :Nick.lang-LFIN    :cmd.EXEC AP2MREN AP2LFIN APL2LANG &-FM SUOMI ;
|  :Nick.lang-LFRA    :cmd.EXEC AP2MREN AP2LFRA APL2LANG &-FM FRANCAIS ;
|  :Nick.lang-LFRC    :cmd.EXEC AP2MREN AP2LFRC APL2LANG &-FM FRC ;
|  :Nick.lang-LGRE    :cmd.EXEC AP2MREN AP2LGRE APL2LANG &-FM GREETING ;
|  :Nick.lang-LHEB    :cmd.EXEC AP2MREN AP2LHEB APL2LANG &-FM HEBREW ;
|  :Nick.lang-LITA    :cmd.EXEC AP2MREN AP2LITA APL2LANG &-FM ITALIANO ;
|  :Nick.lang-LJAP    :cmd.EXEC AP2MREN AP2LJAP APL2LANG &-FM JAPANESE ;
|  :Nick.lang-LKAT    :cmd.EXEC AP2MREN AP2LKAT APL2LANG &-FM KATAKANA ;
|  :Nick.lang-LMAG    :cmd.EXEC AP2MREN AP2LMAG APL2LANG &-FM MAGYAR ;
|  :Nick.lang-LNOR    :cmd.EXEC AP2MREN AP2LNOR APL2LANG &-FM NORSK ;
|  :Nick.lang-LPTG    :cmd.EXEC AP2MREN AP2LPTG APL2LANG &-FM PORTUGUE ;
|  :Nick.lang-LSVE    :cmd.EXEC AP2MREN AP2LSVE APL2LANG &-FM SVENSKA ;

|  :Nick.help-HDEU    :cmd.EXEC AP2MREN AP2HDEU APL2HELP &-FM DEUTSCH ;
|  :Nick.help-HENP    :cmd.EXEC AP2MREN AP2HENP APL2HELP &-FM ENP ;
|  :Nick.help-HENU    :cmd.EXEC AP2MREN AP2HENU APL2HELP &-FM DEFAULT ;
|  :Nick.help-HJAP    :cmd.EXEC AP2MREN AP2HJAP APL2HELP &-FM JAPANESE ;

|  :Nick.fp1-CA :cmd.EXEC AP2MREN AP2C55A1 FONT3812 &-FM C08155A0 ;
|  :Nick.fp1-CB :cmd.EXEC AP2MREN AP2C55B1 FONT3812 &-FM C08155B0 ;
|  :Nick.fp1-CD :cmd.EXEC AP2MREN AP2C55D1 FONT3812 &-FM C08155D0 ;
|  :Nick.fp1-CF :cmd.EXEC AP2MREN AP2C55F1 FONT3812 &-FM C08155F0 ;
|  :Nick.fp1-CH :cmd.EXEC AP2MREN AP2C55H1 FONT3812 &-FM C08155H0 ;
```

```
|  :Nick.fp1-CJ :cmd.EXEC AP2MREN AP2C55J1 FONT3812 &-FM C08155J0 ;
|  :Nick.fp1-CN :cmd.EXEC AP2MREN AP2C55N1 FONT3812 &-FM C08155N0 ;
|  :Nick.fp1-C0 :cmd.EXEC AP2MREN AP2C5501 FONT3812 &-FM C0815500 ;
|  :Nick.fp1-C6 :cmd.EXEC AP2MREN AP2C5561 FONT3812 &-FM C0815560 ;
|  :Nick.fp1-C7 :cmd.EXEC AP2MREN AP2C5571 FONT3812 &-FM C0815570 ;
|  :Nick.fp1-C8 :cmd.EXEC AP2MREN AP2C5581 FONT3812 &-FM C0815580 ;
|  :Nick.fp1-C9 :cmd.EXEC AP2MREN AP2C5591 FONT3812 &-FM C0815590 ;

|  :Nick.fp2-CA :cmd.EXEC AP2MREN AP2C55A2 FONT3820 &-FM C08155A0 ;
|  :Nick.fp2-CB :cmd.EXEC AP2MREN AP2C55B2 FONT3820 &-FM C08155B0 ;
|  :Nick.fp2-CD :cmd.EXEC AP2MREN AP2C55D2 FONT3820 &-FM C08155D0 ;
|  :Nick.fp2-CF :cmd.EXEC AP2MREN AP2C55F2 FONT3820 &-FM C08155F0 ;
|  :Nick.fp2-CH :cmd.EXEC AP2MREN AP2C55H2 FONT3820 &-FM C08155H0 ;
|  :Nick.fp2-CJ :cmd.EXEC AP2MREN AP2C55J2 FONT3820 &-FM C08155J0 ;
|  :Nick.fp2-CN :cmd.EXEC AP2MREN AP2C55N2 FONT3820 &-FM C08155N0 ;
|  :Nick.fp2-C0 :cmd.EXEC AP2MREN AP2C5502 FONT3820 &-FM C0815500 ;
|  :Nick.fp2-C6 :cmd.EXEC AP2MREN AP2C5562 FONT3820 &-FM C0815560 ;
|  :Nick.fp2-C7 :cmd.EXEC AP2MREN AP2C5572 FONT3820 &-FM C0815570 ;
|  :Nick.fp2-C8 :cmd.EXEC AP2MREN AP2C5582 FONT3820 &-FM C0815580 ;
|  :Nick.fp2-C9 :cmd.EXEC AP2MREN AP2C5592 FONT3820 &-FM C0815590 ;
|  :Nick.fp2-T3 :cmd.EXEC AP2MREN AP2T2932 FONT3820 &-FM T1200293 ;
|  :Nick.fp2-XA :cmd.EXEC AP2MREN AP2X55A2 FONT3820 &-FM X08155AB ;
|  :Nick.fp2-XB :cmd.EXEC AP2MREN AP2X55B2 FONT3820 &-FM X08155BB ;
|  :Nick.fp2-XD :cmd.EXEC AP2MREN AP2X55D2 FONT3820 &-FM X08155DB ;
|  :Nick.fp2-XF :cmd.EXEC AP2MREN AP2X55F2 FONT3820 &-FM X08155FB ;
|  :Nick.fp2-XH :cmd.EXEC AP2MREN AP2X55H2 FONT3820 &-FM X08155HB ;
|  :Nick.fp2-XJ :cmd.EXEC AP2MREN AP2X55J2 FONT3820 &-FM X08155JB ;
|  :Nick.fp2-XN :cmd.EXEC AP2MREN AP2X55N2 FONT3820 &-FM X08155NB ;
|  :Nick.fp2-X0 :cmd.EXEC AP2MREN AP2X5502 FONT3820 &-FM X081550B ;
|  :Nick.fp2-X6 :cmd.EXEC AP2MREN AP2X5562 FONT3820 &-FM X081556B ;
|  :Nick.fp2-X7 :cmd.EXEC AP2MREN AP2X5572 FONT3820 &-FM X081557B ;
|  :Nick.fp2-X8 :cmd.EXEC AP2MREN AP2X5582 FONT3820 &-FM X081558B ;
|  :Nick.fp2-X9 :cmd.EXEC AP2MREN AP2X5592 FONT3820 &-FM X081559B ;

|  :Nick.fpp-CA :cmd.EXEC AP2MREN AP2C55AP FONT38PP &-FM C18155A0 ;
|  :Nick.fpp-CB :cmd.EXEC AP2MREN AP2C55BP FONT38PP &-FM C18155B0 ;
|  :Nick.fpp-CD :cmd.EXEC AP2MREN AP2C55DP FONT38PP &-FM C18155D0 ;
|  :Nick.fpp-CF :cmd.EXEC AP2MREN AP2C55FP FONT38PP &-FM C18155F0 ;
|  :Nick.fpp-CH :cmd.EXEC AP2MREN AP2C55HP FONT38PP &-FM C18155H0 ;
|  :Nick.fpp-CJ :cmd.EXEC AP2MREN AP2C55JP FONT38PP &-FM C18155J0 ;
|  :Nick.fpp-CN :cmd.EXEC AP2MREN AP2C55NP FONT38PP &-FM C18155N0 ;
|  :Nick.fpp-C0 :cmd.EXEC AP2MREN AP2C550P FONT38PP &-FM C1815500 ;
|  :Nick.fpp-C6 :cmd.EXEC AP2MREN AP2C556P FONT38PP &-FM C1815560 ;
|  :Nick.fpp-C7 :cmd.EXEC AP2MREN AP2C557P FONT38PP &-FM C1815570 ;
|  :Nick.fpp-C8 :cmd.EXEC AP2MREN AP2C558P FONT38PP &-FM C1815580 ;
|  :Nick.fpp-C9 :cmd.EXEC AP2MREN AP2C559P FONT38PP &-FM C1815590 ;
|  :Nick.fpp-T3 :cmd.EXEC AP2MREN AP2T293P FONT38PP &-FM T1200293 ;
|  :Nick.fpp-XA :cmd.EXEC AP2MREN AP2X55AP FONT38PP &-FM X18155AB ;
|  :Nick.fpp-XB :cmd.EXEC AP2MREN AP2X55BP FONT38PP &-FM X18155BB ;
|  :Nick.fpp-XD :cmd.EXEC AP2MREN AP2X55DP FONT38PP &-FM X18155DB ;
|  :Nick.fpp-XF :cmd.EXEC AP2MREN AP2X55FP FONT38PP &-FM X18155FB ;
|  :Nick.fpp-XH :cmd.EXEC AP2MREN AP2X55HP FONT38PP &-FM X18155HB ;
|  :Nick.fpp-XJ :cmd.EXEC AP2MREN AP2X55JP FONT38PP &-FM X18155JB ;
|  :Nick.fpp-XN :cmd.EXEC AP2MREN AP2X55NP FONT38PP &-FM X18155NB ;
```

```
:Nick.fpp-X0 :cmd.EXEC AP2MREN AP2X550P FONT38PP &-FM X181550B ;
:Nick.fpp-X6 :cmd.EXEC AP2MREN AP2X556P FONT38PP &-FM X181556B ;
:Nick.fpp-X7 :cmd.EXEC AP2MREN AP2X557P FONT38PP &-FM X181557B ;
:Nick.fpp-X8 :cmd.EXEC AP2MREN AP2X558P FONT38PP &-FM X181558B ;
:Nick.fpp-X9 :cmd.EXEC AP2MREN AP2X559P FONT38PP &-FM X181559B ;

:rem.*-----------------------------------------------------------------*
*
*  Macros
*
*-----------------------------------------------------------------*

:Nick.APL2-MAC  :rem.Update the maclibs supplied with APL2
                :list. vsa-MACS RC28=0 ap2-MACS

:Nick.ap2-MACS  :rem.Update the AP2MAC MACLIB
                :cmd.CMDCALL STATE AP2MAC MACLIB *
                :list.a2mcopy a2m-1 a2m-2 a2m-3 a2m-4 RC28=0 a2mcomp
  :Nick.a2m-1     :list. A2MVOPT A2MTAG A2MRESAP A2MCHECK
  :Nick.a2m-2     :list. A2MVPTV A2MPTH A2MXPTX  A2MXVCT
  :Nick.a2m-3     :list. A2MGSPC A2MFSPC
  :Nick.a2m-4     :list. A2MSDF A2MCDR A2MECV A2MSCV A2MPCV A2MCSVPE
  :Nick.a2mcopy :rem.Ensure maclib is on target disk
                :cmd.&CALL -<COPY> AP2MAC MACLIB * = = &-FM OLDD REPL
  :Nick.a2mcomp :cmd.CMDCALL MACLIB COMP AP2MAC
     :Nick.A2MVOPT      :cmd. STATE AP2VOPT  COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2VOPT
     :Nick.A2MTAG       :cmd. STATE AP2TAG   COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2TAG
     :Nick.A2MRESAP     :cmd. STATE AP2RESAP COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2RESAP
     :Nick.A2MVPTV      :cmd. STATE AP2VPTV  COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2VPTV
     :Nick.A2MPTH       :cmd. STATE AP2PTH   COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2PTH
     :Nick.A2MXPTX      :cmd. STATE AP2XPTX  COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2XPTX
     :Nick.A2MXVCT      :cmd. STATE AP2XVCT  COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2XVCT
     :Nick.A2MCHECK     :cmd. STATE AP2CHECK COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2CHECK
     :Nick.A2MGSPC      :cmd. STATE AP2GSPC  COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2GSPC
     :Nick.A2MFSPC      :cmd. STATE AP2FSPC  COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2FSPC
     :Nick.A2MSDF       :cmd. STATE AP2SDF   COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2SDF
     :Nick.A2MCDR       :cmd. STATE AP2CDR   COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2CDR
     :Nick.A2MECV       :cmd. STATE AP2ECV   COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2ECV
     :Nick.A2MSCV       :cmd. STATE AP2SCV   COPY ;
                              CMDCALL MACLIB REP AP2MAC AP2SCV
     :Nick.A2MPCV       :cmd. STATE AP2PCV   COPY ;
```

```
|                                CMDCALL MACLIB REP AP2MAC AP2PCV
|       :Nick.A2MCSVPE     :cmd. STATE AP2CSVPE COPY ;
|                                CMDCALL MACLIB REP AP2MAC AP2CSVPE

|  :Nick.vsa-MACS  :rem.Update the AP2VSAPL MACLIB if present
|                  :cmd. STATE AP2VSAPL MACLIB *
|    :list.vsmcopy VSMAC-1 VSMAC-2 VSMAC-3 VSMAC-4 VSMACOMP
|    :Nick.VSMAC-1  :list. VSMDESC VSMKSTOZ VSMKZTOS VSMPCV VSMREGS
|    :Nick.VSMAC-2  :list. VSMSCV VSMSHSVP VSMZCODE VSMDFORM VSMPACC
|    :Nick.VSMAC-3  :list. VSMPCPY VSMPOFR VSMPQRY VSMPREF VSMPRET
|    :Nick.VSMAC-4  :list. VSMPSOF VSMPSON VSMPSPC VSMPWAIT VSMSDF VSMCDR
|    :Nick.vsmcopy :rem.Ensure maclib is on target disk
|                  :cmd.&CALL -<COPY> AP2VSAPL MACLIB * = = &-FM OLDD REPL
|    :Nick.VSMACOMP :cmd. CMDCALL MACLIB COMP AP2VSAPL
|       :Nick.VSMDESC      :cmd. STATE APLDESC  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL APLDESC
|       :Nick.VSMKSTOZ     :cmd. STATE APLKSTOZ COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL APLKSTOZ
|       :Nick.VSMKZTOS     :cmd. STATE APLKZTOS COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL APLKZTOS
|       :Nick.VSMPCV       :cmd. STATE APLPCV   COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL APLPCV
|       :Nick.VSMREGS      :cmd. STATE APLREGS  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL APLREGS
|       :Nick.VSMSCV       :cmd. STATE APLSCV   COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL APLSCV
|       :Nick.VSMSHSVP     :cmd. STATE APLSHSVP COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL APLSHSVP
|       :Nick.VSMZCODE     :cmd. STATE APLZCODE COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL APLZCODE
|       :Nick.VSMDFORM     :cmd. STATE ASVDFORM COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVDFORM
|       :Nick.VSMPACC      :cmd. STATE ASVPACC  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVPACC
|       :Nick.VSMPCPY      :cmd. STATE ASVPCPY  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVPCPY
|       :Nick.VSMPOFR      :cmd. STATE ASVPOFR  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVPOFR
|       :Nick.VSMPQRY      :cmd. STATE ASVPQRY  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVPQRY
|       :Nick.VSMPREF      :cmd. STATE ASVPREF  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVPREF
|       :Nick.VSMPRET      :cmd. STATE ASVPRET  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVPRET
|       :Nick.VSMPSOF      :cmd. STATE ASVPSOF  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVPSOF
|       :Nick.VSMPSON      :cmd. STATE ASVPSON  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVPSON
|       :Nick.VSMPSPC      :cmd. STATE ASVPSPC  COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVPSPC
|       :Nick.VSMPWAIT     :cmd. STATE ASVPWAIT COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL ASVPWAIT
|       :Nick.VSMSDF       :cmd. STATE AP2SDF   COPY ;
|                                CMDCALL MACLIB REP AP2VSAPL AP2SDF
```

```
|      :Nick.VSMCDR       :cmd. STATE AP2CDR   COPY ;
|                               CMDCALL MACLIB REP AP2VSAPL AP2CDR

| :rem.*-------------------------------------------------------------*
| *
| *  Workspaces - Maintained by AP2MWS (corrective service only).
| *               :Nick WS-Build calls AP2MWS to rebuild workspaces
| *               from APLIMAGE files.
| *
| *-------------------------------------------------------------*

| :Nick.WS-Build   :cmd.FILEDEF AP2XLOAD DISK APL2 LOADLIB * ;
|                         AP2XLOAD APL2 APL2 AP2XLOAD
|                    :msg.*** WORKSPACE MAINTENANCE ( &-GDDM &-SQL )
|             :list.WS-Password iw-Base iw-&-GDDM iw-&-SQL WS-DROP RC28=0

|   :Nick.WS-Password  :cmd.&-Password = ALL
|   :Nick.WS-DROP      :cmd.NUCXDROP APL2
|   :Nick.iw-Base      :list.iwB-1 iwB-2 iwB-3
|   :Nick.iwB-1        :list.w-APLDATA  w-CHARTX   w-CMS      w-DISPLAY
|   :Nick.iwB-2        :list.w-EXAMPLES w-MATHFNS  w-MEDIT    w-PRINTWS
|   :Nick.iwB-3        :list.w-TRANSFER w-UTILITY  w-VAPLFILE w-VSAMDATA
|                           w-WSINFO   w-SUPPLIED

|   :Nick.iw-GDDM      :list.w-FSC126   w-FSM      w-GDMX     w-GRAPHPAK
|   :Nick.iw-SQL       :list.w-SQL
|   :Nick.iw-NOGDDM    :rem.* Placeholder
|   :Nick.iw-NOSQL     :rem.* Placeholder

| :Nick.w-APLDATA  :cmd.EXEC AP2MWS AP2WAPLD 2 APLDATA  &-Password (&-Sys
| :Nick.w-CHARTX   :cmd.EXEC AP2MWS AP2WCHAR 2 CHARTX   &-Password (&-Sys
| :Nick.w-CMS      :cmd.EXEC AP2MWS AP2WCMS  2 CMS      &-Password (&-Sys
| :Nick.w-DISPLAY  :cmd.EXEC AP2MWS AP2WDISP 1 DISPLAY  &-Password (&-Sys
| :Nick.w-EXAMPLES :cmd.EXEC AP2MWS AP2WEXAM 1 EXAMPLES &-Password (&-Sys
| :Nick.w-FSC126   :cmd.EXEC AP2MWS AP2WFSC1 2 FSC126   &-Password (&-Sys
| :Nick.w-FSM      :cmd.EXEC AP2MWS AP2WFSM  2 FSM      &-Password (&-Sys
| :Nick.w-GDMX     :cmd.EXEC AP2MWS AP2WGDMX 2 GDMX     &-Password (&-Sys
| :Nick.w-GRAPHPAK :cmd.EXEC AP2MWS AP2WGRAP 2 GRAPHPAK &-Password (&-Sys
| :Nick.w-MATHFNS  :cmd.EXEC AP2MWS AP2WMATH 1 MATHFNS  &-Password (&-Sys
| :Nick.w-MEDIT    :cmd.EXEC AP2MWS AP2WMEDI 1 MEDIT    &-Password (&-Sys
| :Nick.w-PRINTWS  :cmd.EXEC AP2MWS AP2WPRIN 2 PRINTWS  &-Password (&-Sys
| :Nick.w-SQL      :cmd.EXEC AP2MWS AP2WSQL  2 SQL      &-Password (&-Sys
| :Nick.w-SUPPLIED :cmd.EXEC AP2MWS AP2WSUPP 1 SUPPLIED &-Password (&-Sys
| :Nick.w-TRANSFER :cmd.EXEC AP2MWS AP2WTRAN 2 TRANSFER &-Password (&-Sys
| :Nick.w-UTILITY  :cmd.EXEC AP2MWS AP2WUTIL 1 UTILITY  &-Password (&-Sys
| :Nick.w-VAPLFILE :cmd.EXEC AP2MWS AP2WVAPL 2 VAPLFILE &-Password (&-Sys
| :Nick.w-VSAMDATA :cmd.EXEC AP2MWS AP2WVSAM 2 VSAMDATA &-Password (&-Sys
| :Nick.w-WSINFO   :cmd.EXEC AP2MWS AP2WWSIN 1 WSINFO   &-Password (&-Sys

| :rem.*-------------------------------------------------------------*
| *
| *  AP2MLNK tags - The Visible Code Objects
| *
| *    1) For AP2MLNK ____ MODULE ... or AP2MLNK ___ LOADLIB ...,
```

```
| *           The appropriate MODULE-___ or LOADLIB-___ tag is used.
| *           This contains the original definition of the object
| *           prior to any customizing.
| *
| *    2) When AP2MLNK processes 'part' tag:
| *           a. If second word absent, searches
| *              assuming filetype specfied by LOADLIB-TEXTFT
| *              or MODULE-TEXTFT, whichever is appropriate
| *           b. If no text is found but a base module will
| *              later be 'INCLUDEd' from the SOURCE library,
| *              an 'ORDER' control card is created.
| *
| *--------------------------------------------------------------------*

| :Nick.LOADLIB-APL2           :lked. ENTRY AP2VIAAA
|                              :list. p-BASE
| :Nick.LOADLIB-APL2AE         :lked. ENTRY AP2VIAAA
|                              :list. p-BASE
| :Nick.LOADLIB-AP2HIGH        :lked. ENTRY AP2VIAAA
|                              :list. p-Language
| :Nick.LOADLIB-AP2HIAE        :lked. ENTRY AP2VIAAA
|                              :list. p-Language
| :Nick.MODULE-APL2            :list. VIAPL VIOPT
| :Nick.MODULE-APL2AE          :list. VIAPL VIOPE

| :Nick.LOADLIB-AP2CSVP        :list. INCLUDE VIAAA c-SVP VIZZZ
| :Nick.MODULE-AP2CSVP         :list. VCSVS

| :Nick.MODULE-AP2XLOAD        :list. VMRLD
| :Nick.MODULE-AP2VMCF         :list. VMEXT
| :Nick.MODULE-AP2VUTIL        :list. VIDCS
| :Nick.MODULE-APL2PI
|   :cmd.&IF .&CP NE .VM/SP &GOPTS = &STRING OF &GOPTS AMODE 31 RMODE 24
|                              :list. VAPI XAPIC

| :Nick.LOADLIB-AP2VNRP1       :cmd.&-TEXTFT = &STRING OF TEXT TXT&-LV
|                              :lked. ENTRY AP2PDDT
|                              :list. INCLUDE PDDT

| :Nick.LOADLIB-AP2VNRP2       :cmd.&-TEXTFT = &STRING OF TEXT TXT&-LV
|                              :lked. ENTRY AP2PPKG
|                              :list. INCLUDE PPKG

| :Nick.LOADLIB-AP2VNRP3       :cmd.&-TEXTFT = &STRING OF TEXT TXT&-LV
|                              :lked. ENTRY AP2PDFMT
|                              :list. INCLUDE PDFMT

| :Nick.LOADLIB-AP2VNRP4       :cmd.&-TEXTFT = &STRING OF TEXT TXT&-LV
|                              :lked. ENTRY AP2PIVP
|                              :list. INCLUDE PIVP

| :Nick.LOADLIB-AP2VNRP5       :cmd.&-TEXTFT = &STRING OF TEXT TXT&-LV
|                              :lked. ENTRY AP2PAPIW
|                              :list. INCLUDE PAPIW
```

```
| :Nick.LOADLIB-AP2VNRPS       :cmd.&-TEXTFT = &STRING OF TEXT TXT&-LV
|                              :lked. ENTRY AP2PPSRV
|                              :list. INCLUDE PPSRV

| :Nick.LOADLIB-AP2VNRPI       :cmd.&-TEXTFT = &STRING OF TEXT TXT&-LV
|                              :lked. ENTRY AP2PIDI
|                              :list. INCLUDE PIDI

| :Nick.LOADLIB-AP2VNRPR       :cmd.&-TEXTFT = &STRING OF TEXT TXT&-LV
|                              :lked. ENTRY AP2PRAPL
|                              :list. INCLUDE PRAPL

| :Nick.MODULE-TEXTFT  :cmd.&-TEXTFT = &STRING OF TEXT TXT&-LV
| :Nick.LOADLIB-TEXTFT :cmd.&-TEXTFT = &STRING OF TEXT TXT&-LV

| :Nick.Local-TEXT   :list. Local-OPT
| :Nick.Local-AETEXT :list. Local-VIOPE
| :Nick.Local-HIGHTEXT :list. Local-OPT Local-V127I
| :Nick.Local-HIAETEXT :list. Local-VIOPE Local-V127I

| :Nick.Local-OPT    :cmd.STATE AP2VIOPT TEXT *
|                    :part.AP2VIOPT  TEXT
|                    :msg.*** INCLUDE LOCAL APL2 OPTIONS (AP2VIOPT)
| :Nick.Local-VIOPE  :cmd.STATE AP2VIOPE TEXT *
|                    :part.AP2VIOPE  TEXT
|                    :msg.*** INCLUDE LOCAL APL2AE OPTIONS (AP2VIOPE)
| :Nick.Local-V127I  :cmd.STATE AP2V127I TEXT *
|                    :part.AP2V127I  TEXT
|                    :msg.*** INCLUDE LOCAL SQL/DS INTERFACE (AP2V127I)
| :Nick.Local-XAPIC  :cmd.STATE AP2XAPIC TEXT *
|                    :part.AP2XAPIC  TEXT
|                    :msg.*** INCLUDE LOCAL APL2PI INTERFACE (AP2XAPIC)

| :rem.*-----------------------------------------------------------------*
| *
| *  AP2MLNK utility and option tags
| *
| *-----------------------------------------------------------------*

| :Nick.t-APL2SM       :list. INCLUDE c-APL2SM
| :Nick.t-NOAPL2SM    :lked. REPLACE AP2ATYOP,AP2AINIT,AP2ATERM ;
|                            LIBRARY (AP2ATYOP,AP2AINIT,AP2ATERM);
|                            REPLACE AP2AUALT,AP2AUPRO ;
|                            LIBRARY (AP2AUALT,AP2AUPRO)
|                     :msg.*** APL2 SESSION MANAGER SUPPORT REMOVED
|                     :list. EXCLUDE c-APL2SM INCLUDE

| :Nick.t-GDDM         :list. ADMRLIB
| :Nick.t-NOGDDM       :msg.*** GDDM SUPPORT REMOVED
|                      :list. l-NOGDDM
| :Nick.l-NOGDDM       :lked. REPLACE ADMASP     ; LIBRARY (ADMASP)   ;
|                             REPLACE ADMASSNV   ; LIBRARY (ADMASSNV) ;
|                             REPLACE ADMASXV    ; LIBRARY (ADMASXV)  ;
```

```
|                              REPLACE ADMASLV    ; LIBRARY (ADMASLV)

|  :Nick.t-GDMX          :list. INCLUDE c-GDDM
|  :Nick.t-NOGDMX
|                        :list. EXCLUDE c-GDDM  INCLUDE

|  :Nick.t-QMF           :list. DSQCIA  DSQCLOD2 DSQCMCVP
|  :Nick.t-NOQMF
|                        :lked. REPLACE DSQCIA    ; LIBRARY (DSQCIA)
|                               REPLACE DSQCLOD2  ; LIBRARY (DSQCLOD2)
|                               REPLACE DSQCMCVP  ; LIBRARY (DSQCMCVP)

|  :Nick.t-SQL           :list. ARIRVSTC
|  :Nick.t-NOSQL         :msg.*** SQL SUPPORT REMOVED
|                        :lked. REPLACE ARIRVST   ; LIBRARY (ARIRVST)

|  :Nick.t-AP127         :list. INCLUDE c-SQL
|  :Nick.t-NOAP127
|                        :lked. REPLACE AP2X127   ; LIBRARY (AP2X127)
|                        :list. EXCLUDE c-SQL INCLUDE

|  :Nick.o-GDDM3         :lked. ORDER ADMASP,ADMASLV,ADMASSNV,ADMASXV

|  :Nick.ARIRVSTC        :cmd.CMDCALL STATE ARIRVSTC TEXT * ;
|                             FILEDEF ARIRVSTC DISK ARIRVSTC TEXT *
|                                        (RECFM F LRECL 80 BLOCK 80
|                        :lked. INCLUDE ARIRVSTC ;
|                               ORDER ARIRVST
|                                 :msg.*** INCLUDE SQL STUB (ARIRVSTC)

|  :Nick.ADMRLIB         :cmd.CMDCALL STATE ADMRLIB TXTLIB * ;
|                             FILEDEF SYSLIB DISK ADMRLIB TXTLIB *
|                                      (RECFM F LRECL 80 BLOCK 80
|                             :list. c-ADMRLIB1 c-ADMRLIB2
|  :Nick.c-ADMRLIB1             :lked. REPLACE ADMASP     ;
|                                      LIBRARY (ADMASP)   ;
|                                      REPLACE ADMASSNV   ;
|                                      LIBRARY (ADMASSNV) ;
|                                      REPLACE ADMASXV    ;
|                                      LIBRARY (ADMASXV)  ;
|                                      REPLACE ADMASLV    ;
|                                      LIBRARY (ADMASLV)
|  :Nick.c-ADMRLIB2             :lked. INCLUDE SYSLIB(ADMASP) (GDDM STUB) ;
|                                      INCLUDE SYSLIB(ADMASSNV) ;
|                                      INCLUDE SYSLIB(ADMASXV)  ;
|                                      INCLUDE SYSLIB(ADMASLV)
|                                :msg.*** INCLUDE GDDM STUB (ADMASP)
|  :Nick.DSQCIA       :cmd.CMDCALL STATE DSQCIA TEXT * ;
|                          FILEDEF DSQCIA DISK DSQCIA TEXT *
|                                     (RECFM F LRECL 80 BLOCK 80
|                     :lked. INCLUDE DSQCIA ;
|                            ORDER DSQCIA
|                     :msg.*** INCLUDE QMF STUB (DSQCIA)
```

```
| :Nick.DSQCLOD2     :cmd.CMDCALL STATE DSQCLOD2 TEXT * ;
|                         FILEDEF DSQCLOD2 DISK DSQCLOD2 TEXT *
|                                    (RECFM F LRECL 80 BLOCK 80
|                    :lked. INCLUDE DSQCLOD2 ;
|                           ORDER DSQCLOD2
|                    :msg.*** INCLUDE QMF STUB (DSQCLOD2)

| :Nick.DSQCMCVP     :cmd.CMDCALL STATE DSQCMCVP TEXT * ;
|                         FILEDEF DSQCMCVP DISK DSQCMCVP TEXT *
|                                    (RECFM F LRECL 80 BLOCK 80
|                    :lked. INCLUDE DSQCMCVP ;
|                           ORDER DSQCMCVP
|                    :msg.*** INCLUDE QMF STUB (DSQCMCVP)

| :Nick.INCLUDE        :cmd.&-TXT = 1
| :Nick.EXCLUDE        :cmd.&-TXT = 0

| :Nick.LOADLIB-AP2C220        :cmd.&-TEXTFT = &STRING OF TXT&-LV
|                              :list. INCLUDE VIAAA c-SVP VIZZZ
| :Nick.MODULE-AP2V220         :list. VIAPL VIOPT
| :Nick.MODULE-AP2E220         :list. VIAPL VIOPE
| :Nick.LOADLIB-AP2V220        :cmd.&-TEXTFT = &STRING OF TXT&-LV
|                              :list. INCLUDE VIAAA p-BASE c-BASE
|                                     t-GDDM t-GDMX o-GDDM3 VIZZZ
| :Nick.LOADLIB-AP2E220        :cmd.&-TEXTFT = &STRING OF TXT&-LV
|                              :list. INCLUDE VIAAA p-BASE c-AEBASE
|                                     t-GDDM t-GDMX o-GDDM3 VIZZZ
| :Nick.LOADLIB-AP2W220        :cmd.&-TEXTFT = &STRING OF TXT&-LV
|                              :list. INCLUDE VIAAA VIOPT c-Language
|                                     p-Language c-HIX t-QMF
|                                     t-APL2SM c-SVP c-HIAP
|                                     t-AP127 t-SQL VIZZZ
| :Nick.LOADLIB-AP2F220        :cmd.&-TEXTFT = &STRING OF TXT&-LV
|                              :list. INCLUDE VIAAA VIOPE c-AElang
|                                     p-Language c-HIX t-QMF
|                                     c-SVP c-HIAPAE
|                                     t-AP127 t-SQL VIZZZ

| :rem.*-------------------------------------------------------------*
| *
| *  TEXT Decks - Informal Convention for Lists
| *
| *        c- = Component (Order determines order within module)
| *        p- = PAGE  cards
| *
| *-------------------------------------------------------------*

| :Nick.c-BASE         :list. INCLUDE c-Executor
| :Nick.c-AEBASE       :list. INCLUDE c-AEexec
| :Nick.p-BASE         :list. p-Executor
| :Nick.p-X127         :rem. Page cards could go here
| :Nick.c-SVP          :rem. lked. PAGE AP2XCSV
|                      :list. XCST XCSV XCSS XCSI VCSC XDTAB
```

```
| :Nick.p-Executor     :rem. PAGE AP2VYYR,AP2XMSTK

| :Nick.c-Executor     :list. VIOPT c-E1 c-E2 c-E3 c-E4 c-E5 c-E6 c-E7
|                             c-E8 c-E9 c-E10 c-E11 c-E12 c-E13
|                             c-E14 c-E15
|                      :msg.*** INCLUDE APL2 RMODE 24 CMS INTERFACE

| :Nick.c-AEexec       :list. VIOPE c-E1 c-E2 c-E3 c-E4 c-E5 c-E6 c-E7
|                             c-E8 c-E9 c-E10 c-E11 c-E12 c-E13
|   :msg.*** INCLUDE APL2 AE RMODE 24 CMS INTERFACE

|   :Nick.c-E1         :list. VINIT VINH  VINIT VINL  VINC  VINT
|   :Nick.c-E2         :list. VICMD VISCN XIPOP XISCA VGCAT
|   :Nick.c-E3         :list. VYENV VYMSC VYSND VYYR
|   :Nick.c-E4         :list. VACTL VASVP XASRV XMSRV
|   :Nick.c-E5         :list. V100  V101  V102  V110  V111 V123
|   :Nick.c-E6         :list. c-X119 c-X124
|   :Nick.c-E7         :list. VNCAL VN10 VN11 XN11
|   :Nick.c-E8         :list. XMSTK VMEXT VMSRV VMDSP VMVCT
|   :Nick.c-E9         :list. VMAIN XMTRI VMTRC VMDMP
|   :Nick.c-E10        :list. VDTBL XDNUM XDREV XDTRA XDCDR
|   :Nick.c-E11        :list. XFNDX VFILE VFILQ VFPRT VFQUL
|   :Nick.c-E12        :list. VFIDO XFNS  VFNSD
|   :Nick.c-E13        :list. VLWSS VTYPC VTYC1 VTQR
|   :Nick.c-E14        :list. VTYPD VTYPE VTYPA VTYPR XTYRA
|   :Nick.c-E15        :list. VLIB  VSPRO ASCHD

|     :Nick.c-X119       :list. X119 X119D X119T V119
|     :Nick.c-X124       :list. X124 X124A V124G X124M

| :Nick.c-APL2SM       :list. c-G1 c-G2 c-G3 c-G4 c-G5 c-G6
|   :Nick.c-G1         :list. ACCBE ACDSL ACHLP
|   :Nick.c-G2         :list. ACNDP ACOPY ACPRM ACPRO ACQRY
|   :Nick.c-G3         :list. ACQUE ACRCP ACRDA ACRSA ACSF
|   :Nick.c-G4         :list. ACXCM ADISP ADMSG ADTTM AFIND
|   :Nick.c-G5         :list. AINI  ALINE APUT
|   :Nick.c-G6         :list. VSDLY VSPR2

| :Nick.c-GDDM         :list. XGCHC VGDDM XGCOM

| :Nick.c-HIX          :list. c-KERN c-MSGS c-NRS c-VN12
|                      :msg.*** INCLUDE RMODE ANY CMS INTERFACE

|   :Nick.c-KERN       :list. XKDBC XKNUM XKTAB XKTXT

|   :Nick.c-MSGS       :list. XMSGB XMSG XMLAN VSALT

|   :Nick.c-NRS        :list. XNRSS XNRS1 XNRS2 XNRS3 XNRS4
|                             XNHLP XNRSQ VNRSQ VNL

|   :Nick.c-VN12       :list. VN12 XN12 XN12A XN12F

| :Nick.c-SQL          :list. X127M X127O X127P X127D
|                             X127T X127F X127J X127U
```

```
|                             X127R X127S V127E V127I

|  :Nick.c-HIAP   :list.c-HIAPAE X104 X120 ASCHD
|  :Nick.c-HIAPAE :list.VASVP XACAP X121 X126 X211

|  :Nick.p-Language     :rem. PAGE AP2IAGO,AP2IOPS

|  :Nick.c-Language     :list. c-Lang1 c-Lang2 c-Lang3
|                       :msg.*** INCLUDE APL2 INTERPRETER

|  :Nick.c-AElang       :list. c-Lang1 c-Lang3
|    :msg.*** INCLUDE APL2 AE INTERPRETER

|    :Nick.c-Lang1      :list. c-L1  c-L2  c-L3  c-L4  c-L5
|                              c-L6  c-L7  c-L8  c-L9  c-L10
|                              c-L11 c-L12 c-L13 c-L14 c-L15
|                              c-L16 c-L17 c-L18 c-L19 c-L20
|                              c-L21 c-L22 c-L23
|    :Nick.c-Lang2      :list. c-L24
|    :Nick.c-Lang3      :list. c-L25

|    :Nick.c-L1         :list. IAAA IAPL
|    :Nick.c-L2         :list. IAGO IAGP ISSS ISST IOCT ISFS ISFM ISFD
|    :Nick.c-L3         :list. ISFC ISXA ISXB ISXI ISXE ISXZ ICIR ICIC
|    :Nick.c-L4         :list. ISVB ISVI ISVE ISVZ ISSY ISVY
|    :Nick.c-L5         :list. ICSR ICSS ICST ICSU ICVR ICVS ICVE ICVZ
|    :Nick.c-L6         :list. IIDY IIDX IFFS IMLR ICOM ISEL IFNC
|    :Nick.c-L7         :list. ITAK ICAT IBIT IMAP ICPI ICPY
|    :Nick.c-L8         :list. IFF0 IRED IRSS ISCN IPAR
|    :Nick.c-L9         :list. IMRI IDFT IDIE IDNA IDRH IFND
|    :Nick.c-L10        :list. INAM ITOC IEFA IJDT
|    :Nick.c-L11        :list. IROT IMAT IGRD IFFF
|    :Nick.c-L12        :list. IFM1 IFM2 IFMA IFME IFMS
|    :Nick.c-L13        :list. IOPS IITM IEXA IDFS
|    :Nick.c-L14        :list. IPRI IDPY ICFO ITFI ITFO ITFS ITFN IOPT
|    :Nick.c-L15        :list. ISNA ISTV IUMX ISVF ISVS
|    :Nick.c-L16        :list. IBLO IERA IEXP IEXS IEXT IEXU IXAV
|    :Nick.c-L17        :list. IAXS IEQU INIM ININ ITPE IUNI
|    :Nick.c-L18        :list. IDEC IENC IDQR IQNL ISAM ISAN
|    :Nick.c-L19        :list. IINZ IINV IEIG IPZE IPZZ
|    :Nick.c-L20        :list. ITNC ITAB ITDC
|    :Nick.c-L21        :list. ITSS ISYS ISYU
|    :Nick.c-L22        :list. IISF IISG IMCK IAPS
|    :Nick.c-L23        :list. ICWS IVER IPDF IPDD1
|    :Nick.c-L24        :list. IPDD2 IFED IFEE ICPJ ICPK
|    :Nick.c-L25        :list. IZZZ

|  :rem.*-------------------------------------------------------------*
|  *
|  *    APL2 Text Decks in Alphabetical Order
|  *
|  *-------------------------------------------------------------*

|  :Nick.TEXT-list    :list. VIAAA VIZZZ VMRLD VIAPL VIDCS VCSVS VNL
```

```
|                            VAPI  VAPV2 XAPIC XCMAP
|                            PDDT PPKG PDFMT PIVP PAPIW PPSRV PIDI PRAPL
|                            c-Executor c-GDDM c-Language c-HIX
|                            c-APL2SM c-SVP c-SQL c-HIAP

| :Nick.AETEXT-list  :list. VIAAA VIZZZ VMRLD VIAPL VIDCS VCSVS VNL
|                            VAPI  VAPV2 XAPIC XCMAP
|                            PDDT PPKG PDFMT PIVP PAPIW PPSRV
|                            c-AEexec c-GDDM c-AElang c-HIX
|                            c-SVP c-SQL c-HIAPAE

|       :Nick.ACCBE   :part.AP2ACCBE
|       :Nick.ACDSL   :part.AP2ACDSL
|       :Nick.ACHLP   :part.AP2ACHLP
|       :Nick.ACNDP   :part.AP2ACNDP
|       :Nick.ACOPY   :part.AP2ACOPY
|       :Nick.ACPRM   :part.AP2ACPRM
|       :Nick.ACPRO   :part.AP2ACPRO
|       :Nick.ACQRY   :part.AP2ACQRY
|       :Nick.ACQUE   :part.AP2ACQUE
|       :Nick.ACRCP   :part.AP2ACRCP
|       :Nick.ACRDA   :part.AP2ACRDA
|       :Nick.ACRSA   :part.AP2ACRSA
|       :Nick.ACSF    :part.AP2ACSF
|       :Nick.ACXCM   :part.AP2ACXCM
|       :Nick.ADISP   :part.AP2ADISP
|       :Nick.ADMSG   :part.AP2ADMSG
|       :Nick.ADTTM   :part.AP2ADTTM
|       :Nick.AFIND   :part.AP2AFIND
|       :Nick.AINI    :part.AP2AINI
|       :Nick.ALINE   :part.AP2ALINE
|       :Nick.APUT    :part.AP2APUT
|       :Nick.ASCHD   :part.AP2ASCHD

|       :Nick.IAAA    :part.AP2IAAA
|       :Nick.IAGO    :part.AP2IAGO
|       :Nick.IAGP    :part.AP2IAGP
|       :Nick.IAPL    :part.AP2IAPL
|       :Nick.IAPS    :part.AP2IAPS
|       :Nick.IAXS    :part.AP2IAXS
|       :Nick.IBIT    :part.AP2IBIT
|       :Nick.IBLO    :part.AP2IBLO
|       :Nick.ICAT    :part.AP2ICAT
|       :Nick.ICFO    :part.AP2ICFO
|       :Nick.ICIC    :part.AP2ICIC
|       :Nick.ICIR    :part.AP2ICIR
|       :Nick.ICOM    :part.AP2ICOM
|       :Nick.ICPI    :part.AP2ICPI
|       :Nick.ICPJ    :part.AP2ICPJ
|       :Nick.ICPK    :part.AP2ICPK
|       :Nick.ICPY    :part.AP2ICPY
|       :Nick.ICSR    :part.AP2ICSR
|       :Nick.ICSS    :part.AP2ICSS
|       :Nick.ICST    :part.AP2ICST
```

```
|       :Nick.ICSU    :part.AP2ICSU
|       :Nick.ICVE    :part.AP2ICVE
|       :Nick.ICVR    :part.AP2ICVR
|       :Nick.ICVS    :part.AP2ICVS
|       :Nick.ICVZ    :part.AP2ICVZ
|       :Nick.ICWS    :part.AP2ICWS
|       :Nick.IDEC    :part.AP2IDEC
|       :Nick.IDFS    :part.AP2IDFS
|       :Nick.IDFT    :part.AP2IDFT
|       :Nick.IDIE    :part.AP2IDIE
|       :Nick.IDNA    :part.AP2IDNA
|       :Nick.IDPY    :part.AP2IDPY
|       :Nick.IDQR    :part.AP2IDQR
|       :Nick.IDRH    :part.AP2IDRH
|       :Nick.IEFA    :part.AP2IEFA
|       :Nick.IEIG    :part.AP2IEIG
|       :Nick.IENC    :part.AP2IENC
|       :Nick.IEQU    :part.AP2IEQU
|       :Nick.IERA    :part.AP2IERA
|       :Nick.IEXA    :part.AP2IEXA
|       :Nick.IEXP    :part.AP2IEXP
|       :Nick.IEXS    :part.AP2IEXS
|       :Nick.IEXT    :part.AP2IEXT
|       :Nick.IEXU    :part.AP2IEXU
|       :Nick.IFED    :part.AP2IFED
|       :Nick.IFEE    :part.AP2IFEE
|       :Nick.IFFF    :part.AP2IFFF
|       :Nick.IFFS    :part.AP2IFFS
|       :Nick.IFF0    :part.AP2IFF0
|       :Nick.IFMA    :part.AP2IFMA
|       :Nick.IFME    :part.AP2IFME
|       :Nick.IFMS    :part.AP2IFMS
|       :Nick.IFM1    :part.AP2IFM1
|       :Nick.IFM2    :part.AP2IFM2
|       :Nick.IFNC    :part.AP2IFNC
|       :Nick.IFND    :part.AP2IFND
|       :Nick.IGRD    :part.AP2IGRD
|       :Nick.IIDX    :part.AP2IIDX
|       :Nick.IIDY    :part.AP2IIDY
|       :Nick.IINV    :part.AP2IINV
|       :Nick.IINZ    :part.AP2IINZ
|       :Nick.IISF    :part.AP2IISF
|       :Nick.IISG    :part.AP2IISG
|       :Nick.IITM    :part.AP2IITM
|       :Nick.IJDT    :part.AP2IJDT
|       :Nick.IMAP    :part.AP2IMAP
|       :Nick.IMAT    :part.AP2IMAT
|       :Nick.IMCK    :part.AP2IMCK
|       :Nick.IMLR    :part.AP2IMLR
|       :Nick.IMRI    :part.AP2IMRI
|       :Nick.INAM    :part.AP2INAM
|       :Nick.INIM    :part.AP2INIM
|       :Nick.ININ    :part.AP2ININ
|       :Nick.IOCT    :part.AP2IOCT
```

```
|       :Nick.IOPS    :part.AP2IOPS
|       :Nick.IOPT    :part.AP2IOPT
|       :Nick.IPAR    :part.AP2IPAR
|       :Nick.IPDD1   :part.AP2IPDD1
|       :Nick.IPDD2   :part.AP2IPDD2
|       :Nick.IPDF    :part.AP2IPDF
|       :Nick.IPRI    :part.AP2IPRI
|       :Nick.IPZE    :part.AP2IPZE
|       :Nick.IPZZ    :part.AP2IPZZ
|       :Nick.IQNL    :part.AP2IQNL
|       :Nick.IRED    :part.AP2IRED
|       :Nick.IROT    :part.AP2IROT
|       :Nick.IRSS    :part.AP2IRSS
|       :Nick.ISAM    :part.AP2ISAM
|       :Nick.ISAN    :part.AP2ISAN
|       :Nick.ISCN    :part.AP2ISCN
|       :Nick.ISEL    :part.AP2ISEL
|       :Nick.ISFC    :part.AP2ISFC
|       :Nick.ISFD    :part.AP2ISFD
|       :Nick.ISFM    :part.AP2ISFM
|       :Nick.ISFS    :part.AP2ISFS
|       :Nick.ISNA    :part.AP2ISNA
|       :Nick.ISSS    :part.AP2ISSS
|       :Nick.ISST    :part.AP2ISST
|       :Nick.ISSY    :part.AP2ISSY
|       :Nick.ISTV    :part.AP2ISTV
|       :Nick.ISVB    :part.AP2ISVB
|       :Nick.ISVE    :part.AP2ISVE
|       :Nick.ISVI    :part.AP2ISVI
|       :Nick.ISVF    :part.AP2ISVF
|       :Nick.ISVS    :part.AP2ISVS
|       :Nick.ISVY    :part.AP2ISVY
|       :Nick.ISVZ    :part.AP2ISVZ
|       :Nick.ISXA    :part.AP2ISXA
|       :Nick.ISXB    :part.AP2ISXB
|       :Nick.ISXE    :part.AP2ISXE
|       :Nick.ISXI    :part.AP2ISXI
|       :Nick.ISXZ    :part.AP2ISXZ
|       :Nick.ISYS    :part.AP2ISYS
|       :Nick.ISYU    :part.AP2ISYU
|       :Nick.ITAB    :part.AP2ITAB
|       :Nick.ITAK    :part.AP2ITAK
|       :Nick.ITDC    :part.AP2ITDC
|       :Nick.ITFI    :part.AP2ITFI
|       :Nick.ITFN    :part.AP2ITFN
|       :Nick.ITFO    :part.AP2ITFO
|       :Nick.ITFS    :part.AP2ITFS
|       :Nick.ITNC    :part.AP2ITNC
|       :Nick.ITOC    :part.AP2ITOC
|       :Nick.ITPE    :part.AP2ITPE
|       :Nick.ITSS    :part.AP2ITSS
|       :Nick.IUMX    :part.AP2IUMX
|       :Nick.IUNI    :part.AP2IUNI
|       :Nick.IVER    :part.AP2IVER
```

```
|       :Nick.IXAV    :part.AP2IXAV
|       :Nick.IZZZ    :part.AP2IZZZ

|       :Nick.PAPIW   :part.AP2PAPIW
|       :Nick.PDDT    :part.AP2PDDT
|       :Nick.PDFMT   :part.AP2PDFMT
|       :Nick.PIDI    :part.AP2PIDI
|       :Nick.PIVP    :part.AP2PIVP
|       :Nick.PPKG    :part.AP2PPKG
|       :Nick.PPSRV   :part.AP2PPSRV
|       :Nick.PRAPL   :part.AP2PRAPL

|       :Nick.VACTL   :part.AP2VACTL
|       :Nick.VAPI    :part.AP2VAPI
|       :Nick.VAPV2   :part.AP2VAPV2
|       :Nick.VASVP   :part.AP2VASVP
|       :Nick.VCSC    :part.AP2VCSC
|       :Nick.VCSVS   :part.AP2VCSVS
|       :Nick.VDTBL   :part.AP2VDTBL
|       :Nick.VFIDO   :part.AP2VFIDO
|       :Nick.VFILE   :part.AP2VFILE
|       :Nick.VFILQ   :part.AP2VFILQ
|       :Nick.VFNSD   :part.AP2VFNSD
|       :Nick.VFPRT   :part.AP2VFPRT
|       :Nick.VFQUL   :part.AP2VFQUL
|       :Nick.VGCAT   :part.AP2VGCAT
|       :Nick.VGDDM   :part.AP2VGDDM
|       :Nick.VIAAA   :part.AP2VIAAA
|                     :lked.ENTRY AP2VIAAA
|       :Nick.VIAPL   :part.AP2VIAPL
|       :Nick.VICMD   :part.AP2VICMD
|       :Nick.VIDCS   :part.AP2VIDCS
|       :Nick.VINC    :part.AP2VINC
|       :Nick.VINH    :part.AP2VINH
|       :Nick.VINIT   :part.AP2VINIT
|       :Nick.VINL    :part.AP2VINL
|       :Nick.VINT    :part.AP2VINT
|       :Nick.VIOPE   :part.AP2VIOPE
|       :Nick.VIOPT   :part.AP2VIOPT
|       :Nick.VISCN   :part.AP2VISCN
|       :Nick.VIZZZ   :part.AP2VIZZZ
|       :Nick.VLIB    :part.AP2VLIB
|       :Nick.VLWSS   :part.AP2VLWSS
|       :Nick.VMAIN   :part.AP2VMAIN
|       :Nick.VMDMP   :part.AP2VMDMP
|       :Nick.VMDSP   :part.AP2VMDSP
|       :Nick.VMEXT   :part.AP2VMEXT
|       :Nick.VMRLD   :part.AP2VMRLD
|       :Nick.VMSRV   :part.AP2VMSRV
|       :Nick.VMTRC   :part.AP2VMTRC
|       :Nick.VMVCT   :part.AP2VMVCT
|       :Nick.VNCAL   :part.AP2VNCAL
|       :Nick.VNL     :part.AP2VNL
|       :Nick.VNRSQ   :part.AP2VNRSQ
```

```
|       :Nick.VN10    :part.AP2VN10
|       :Nick.VN11    :part.AP2VN11
|       :Nick.VN12    :part.AP2VN12
|       :Nick.VSALT   :part.AP2VSALT
|       :Nick.VSDLY   :part.AP2VSDLY
|       :Nick.VSPRO   :part.AP2VSPRO
|       :Nick.VSPR2   :part.AP2VSPR2
|       :Nick.VTQR    :part.AP2VTQR
|       :Nick.VTYC1   :part.AP2VTYC1
|       :Nick.VTYPA   :part.AP2VTYPA
|       :Nick.VTYPC   :part.AP2VTYPC
|       :Nick.VTYPD   :part.AP2VTYPD
|       :Nick.VTYPE   :part.AP2VTYPE
|       :Nick.VTYPR   :part.AP2VTYPR
|       :Nick.VYENV   :part.AP2VYENV
|       :Nick.VYMSC   :part.AP2VYMSC
|       :Nick.VYSND   :part.AP2VYSND
|       :Nick.VYYR    :part.AP2VYYR
|       :Nick.V100    :part.AP2V100
|       :Nick.V101    :part.AP2V101
|       :Nick.V102    :part.AP2V102
|       :Nick.V110    :part.AP2V110
|       :Nick.V111    :part.AP2V111
|       :Nick.V119    :part.AP2V119
|       :Nick.V123    :part.AP2V123
|       :Nick.V124G   :part.AP2V124G
|       :Nick.V127E   :part.AP2V127E
|       :Nick.V127I   :part.AP2V127I
|       :Nick.XACAP   :part.AP2XACAP
|       :Nick.XAPIC   :part.AP2XAPIC
|       :Nick.XASRV   :part.AP2XASRV
|       :Nick.XCMAP   :part.AP2XCMAP
|       :Nick.XCSI    :part.AP2XCSI
|       :Nick.XCSS    :part.AP2XCSS
|       :Nick.XCST    :part.AP2XCST
|       :Nick.XCSV    :part.AP2XCSV
|       :Nick.XDCDR   :part.AP2XDCDR
|       :Nick.XDNUM   :part.AP2XDNUM
|       :Nick.XDREV   :part.AP2XDREV
|       :Nick.XDTAB   :part.AP2XDTAB
|       :Nick.XDTRA   :part.AP2XDTRA
|       :Nick.XFNDX   :part.AP2XFNDX
|       :Nick.XFNS    :part.AP2XFNS
|       :Nick.XGCHC   :part.AP2XGCHC
|       :Nick.XGCOM   :part.AP2XGCOM
|       :Nick.XIPOP   :part.AP2XIPOP
|       :Nick.XISCA   :part.AP2XISCA
|       :Nick.XKDBC   :part.AP2XKDBC
|       :Nick.XKNUM   :part.AP2XKNUM
|       :Nick.XKTAB   :part.AP2XKTAB
|       :Nick.XKTXT   :part.AP2XKTXT
|       :Nick.XMLAN   :part.AP2XMLAN
|       :Nick.XMSG    :part.AP2XMSG
|       :Nick.XMSGB   :part.AP2XMSGB
```

```
|       :Nick.XMSRV   :part.AP2XMSRV
|       :Nick.XMSTK   :part.AP2XMSTK
|       :Nick.XMTRI   :part.AP2XMTRI
|       :Nick.XNHLP   :part.AP2XNHLP
|       :Nick.XNRSS   :part.AP2XNRSS
|       :Nick.XNRSQ   :part.AP2XNRSQ
|       :Nick.XNRS1   :part.AP2XNRS1
|       :Nick.XNRS2   :part.AP2XNRS2
|       :Nick.XNRS3   :part.AP2XNRS3
|       :Nick.XNRS4   :part.AP2XNRS4
|       :Nick.XN11    :part.AP2XN11
|       :Nick.XN12    :part.AP2XN12
|       :Nick.XN12A   :part.AP2XN12A
|       :Nick.XN12F   :part.AP2XN12F
|       :Nick.XTYRA   :part.AP2XTYRA
|       :Nick.X104    :part.AP2X104
|       :Nick.X119    :part.AP2X119
|       :Nick.X119D   :part.AP2X119D
|       :Nick.X119T   :part.AP2X119T
|       :Nick.X120    :part.AP2X120
|       :Nick.X121    :part.AP2X121
|       :Nick.X124    :part.AP2X124
|       :Nick.X124A   :part.AP2X124A
|       :Nick.X124M   :part.AP2X124M
|       :Nick.X126    :part.AP2X126
|       :Nick.X127D   :part.AP2X127D
|       :Nick.X127F   :part.AP2X127F
|       :Nick.X127J   :part.AP2X127J
|       :Nick.X127M   :part.AP2X127M
|                     :lked.ENTRY AP2X127
|       :Nick.X127O   :part.AP2X127O
|       :Nick.X127P   :part.AP2X127P
|       :Nick.X127R   :part.AP2X127R
|       :Nick.X127S   :part.AP2X127S
|       :Nick.X127T   :part.AP2X127T
|       :Nick.X127U   :part.AP2X127U
|       :Nick.X211    :part.AP2X211

|  :rem.*-----------------------------------------------------------------*
|  *
|  *  I5688228 Tags - APL2 Tape Definition
|  *
|  *-----------------------------------------------------------------*

|  :Nick.TAPE         :msg.*** SYSTEM APL2 &-LV
|                     :list.File1 t-WTM File2 t-WTM  File3 t-WTM
|                           File4 t-WTM File5 t-WTM  File6 t-WTM
|                           File7 t-WTM File8 t-WTM  File9 t-WTM t-WTM

|  :Nick.File1        :msg.FILE 1 - INSTALL LEVEL and EXEC
|                     :list.F1-INSTLVL  F1-INSTEXEC
|       :Nick.F1-INSTLVL   :part.I5688228 022009
|       :Nick.F1-INSTEXEC  :part.I5688228 EXEC
|  :Nick.File2        :msg.FILE 2 - MEMO TO USERS
```

```
|                    :list.F2-MEMO
|       :Nick.F2-MEMO      :part.I5688228 MEMO
|  :Nick.File3        :msg.FILE 3 - INSTALLATION, SERVICE, and SAMPLE FILES
|                     :list.F3-service F3-AEins F3-samples F3-nrs
|                           F3-lang F3-help F3-prof
|    :Nick.F3-service :list.AEF3-service
|                           F3-MWS F3-VIOPT
|                           F3-V220-m F3-V220-l F3-W220
|       :Nick.F3-MWS       :part.AP2MWS   EXEC
|       :Nick.F3-VIOPT     :part.AP2VIOPT
|       :Nick.F3-V220-l    :part.AP2V220  LOADLIB
|       :Nick.F3-V220-m    :part.AP2V220  MODULE
|       :Nick.F3-W220      :part.AP2W220  LOADLIB
|    :Nick.F3-AEins   :list.AEF1-INSTLVL  AEF1-INSTEXEC  AEF2-MEMO
|    :Nick.F3-samples :list.AEF3-samples F3-VIOPTSAM F3-DEFAULT
|       :Nick.F3-DEFAULT   :part.DEFAULT  AP2MSAMP
|       :Nick.F3-LIBTAB    :part.LIBTAB   AP2MSAMP
|       :Nick.F3-VIOPTSAM  :part.AP2VIOPT AP2MSAMP
|    :Nick.F3-nrs     :list.AEF3-nrs F3-VNRPI F3-VNRPR
|       :Nick.F3-VNRPI     :part.AP2VNRPI LOADLIB
|       :Nick.F3-VNRPR     :part.AP2VNRPR LOADLIB
|    :Nick.F3-lang    :list.F3-LDAN F3-LDEU F3-LENU F3-LESP
|                           F3-LFIN F3-LFRA F3-LFRC F3-LGRE F3-LHEB
|                           F3-LITA F3-LJAP F3-LKAT F3-LMAG F3-LNOR
|                           F3-LPTG F3-LSVE
|       :Nick.F3-LDAN      :part.DANSK    APL2LANG
|       :Nick.F3-LDEU      :part.DEUTSCH  APL2LANG
|       :Nick.F3-LENU      :part.DEFAULT  APL2LANG
|       :Nick.F3-LESP      :part.ESPANOL  APL2LANG
|       :Nick.F3-LFIN      :part.SUOMI    APL2LANG
|       :Nick.F3-LFRA      :part.FRANCAIS APL2LANG
|       :Nick.F3-LFRC      :part.FRC      APL2LANG
|       :Nick.F3-LGRE      :part.GREETING APL2LANG
|       :Nick.F3-LHEB      :part.HEBREW   APL2LANG
|       :Nick.F3-LITA      :part.ITALIANO APL2LANG
|       :Nick.F3-LJAP      :part.JAPANESE APL2LANG
|       :Nick.F3-LKAT      :part.KATAKANA APL2LANG
|       :Nick.F3-LMAG      :part.MAGYAR   APL2LANG
|       :Nick.F3-LNOR      :part.NORSK    APL2LANG
|       :Nick.F3-LPTG      :part.PORTUGUE APL2LANG
|       :Nick.F3-LSVE      :part.SVENSKA  APL2LANG
|    :Nick.F3-help    :list.F3-HDEU F3-HENP F3-HENU F3-HJAP
|       :Nick.F3-HDEU      :part.DEUTSCH  APL2HELP
|       :Nick.F3-HENP      :part.ENP      APL2HELP
|       :Nick.F3-HENU      :part.DEFAULT  APL2HELP
|       :Nick.F3-HJAP      :part.JAPANESE APL2HELP
|    :Nick.F3-prof    :list.F3-TCPIP
|       :Nick.F3-TCPIP     :part.AP2TCPIP APL2PROF
|  :Nick.File4        :msg.FILE 4 - ONLINE HELP
|                     :list.F4-HELP-APL2 F4-HELP-APL2AE
|       :Nick.F4-HELP-APL2   :part.APL2     HELPCMS
|       :Nick.F4-HELP-APL2AE :part.APL2AE   HELPCMS
|  :Nick.File5        :msg.FILE 5 - RUN-TIME AUXILIARY FILES
|                     :list.F5-V5688228 F5-V5688229 F5-AP2VN011
```

```
|                           F5-usertext F5-SSGRAPH F5-SSSM
|       :Nick.F5-V5688228  :part.V5688228 EXEC
|       :Nick.F5-V5688229  :part.V5688229 EXEC
|       :Nick.F5-AP2VN011  :part.AP2VN011 NAMES
|       :Nick.F5-usertext  :list.VNL VAPI VAPV2 XAPIC XCMAP
|       :Nick.F5-SSGRAPH   :list.F5-FVSS F5-1ISSA F5-1ISSC F5-1ISSG
|                               F5-1ISSK F5-1ISSN F5-1ISSR F5-2ISSA
|                               F5-2ISSC F5-2ISSG F5-2ISSK F5-2ISSN
|                               F5-2ISSR
|          :Nick.F5-FVSS   :part.APLFVSS  ADMSYMBL
|          :Nick.F5-1ISSA  :part.APL1ISSA ADMSYMBL
|          :Nick.F5-1ISSC  :part.APL1ISSC ADMSYMBL
|          :Nick.F5-1ISSG  :part.APL1ISSG ADMSYMBL
|          :Nick.F5-1ISSK  :part.APL1ISSK ADMSYMBL
|          :Nick.F5-1ISSN  :part.APL1ISSN ADMSYMBL
|          :Nick.F5-1ISSR  :part.APL1ISSR ADMSYMBL
|          :Nick.F5-2ISSA  :part.APL2ISSA ADMSYMBL
|          :Nick.F5-2ISSC  :part.APL2ISSC ADMSYMBL
|          :Nick.F5-2ISSG  :part.APL2ISSG ADMSYMBL
|          :Nick.F5-2ISSK  :part.APL2ISSK ADMSYMBL
|          :Nick.F5-2ISSN  :part.APL2ISSN ADMSYMBL
|          :Nick.F5-2ISSR  :part.APL2ISSR ADMSYMBL
|       :Nick.F5-SSSM      :part.AP2AISSA ADMSYMBL ;
|                               AP2AISSC ADMSYMBL ;
|                               AP2AISSG ADMSYMBL ;
|                               AP2AISSK ADMSYMBL ;
|                               AP2AISSN ADMSYMBL ;
|                               AP2AISSR ADMSYMBL
|  :Nick.File6        :msg.FILE 6 - WORKSPACES, WS IMAGES
|                     :list.F6-WS F6-IMAGE
|    :Nick.F6-WS      :list.F6-WSLIB1 F6-WSLIB2
|      :Nick.F6-WSLIB1  :list.6-DISPLAY  6-EXAMPLES 6-MATHFNS  6-MEDIT
|                             6-SUPPLIED 6-UTILITY  6-WSINFO
|      :Nick.F6-WSLIB2  :list.6-APLDATA  6-CHARTX   6-CMS      6-FSC126
|                             6-FSM      6-GDMX     6-GRAPHPAK 6-PRINTWS
|                             6-SQL      6-TRANSFER 6-VAPLFILE 6-VSAMDATA
|         :Nick.6-APLDATA    :part.APLDATA  V0000002
|         :Nick.6-CHARTX     :part.CHARTX   V0000002
|         :Nick.6-CMS        :part.CMS      V0000002
|         :Nick.6-DISPLAY    :part.DISPLAY  V0000001
|         :Nick.6-EXAMPLES   :part.EXAMPLES V0000001
|         :Nick.6-FSC126     :part.FSC126   V0000002
|         :Nick.6-FSM        :part.FSM      V0000002
|         :Nick.6-GDMX       :part.GDMX     V0000002
|         :Nick.6-GRAPHPAK   :part.GRAPHPAK V0000002
|         :Nick.6-MATHFNS    :part.MATHFNS  V0000001
|         :Nick.6-MEDIT      :part.MEDIT    V0000001
|         :Nick.6-PRINTWS    :part.PRINTWS  V0000002
|         :Nick.6-SQL        :part.SQL      V0000002
|         :Nick.6-SUPPLIED   :part.SUPPLIED V0000001
|         :Nick.6-TRANSFER   :part.TRANSFER V0000002
|         :Nick.6-UTILITY    :part.UTILITY  V0000001
|         :Nick.6-VAPLFILE   :part.VAPLFILE V0000002
|         :Nick.6-VSAMDATA   :part.VSAMDATA V0000002
```

```
|         :Nick.6-WSINFO     :part.WSINFO   V0000001
|    :Nick.F6-IMAGE   :list.F6-I1 F6-I2 F6-I3 F6-I4
|      :Nick.F6-I1    :list.WAPLD WCHAR WCMS  WDATA WDISP WEXAM WFSC1
|      :Nick.F6-I2    :list.WFSDG WFSM  WGDDM WGDMX WGPVS WGRAP WHOST
|      :Nick.F6-I3    :list.WMATH WMEDI WMESS WPRIN WSQL  WSUPP WTRAN
|      :Nick.F6-I4    :list.WUTIL WVAPL WVSAM WWSIN
|         :Nick.WAPLD        :part.AP2WAPLD APLIMAGE
|         :Nick.WCHAR        :part.AP2WCHAR APLIMAGE
|         :Nick.WCMS         :part.AP2WCMS  APLIMAGE
|         :Nick.WDATA        :part.AP2WDATA APLIMAGE
|         :Nick.WDISP        :part.AP2WDISP APLIMAGE
|         :Nick.WEXAM        :part.AP2WEXAM APLIMAGE
|         :Nick.WFSC1        :part.AP2WFSC1 APLIMAGE
|         :Nick.WFSDG        :part.AP2WFSDG APLIMAGE
|         :Nick.WFSM         :part.AP2WFSM  APLIMAGE
|         :Nick.WGDDM        :part.AP2WGDDM APLIMAGE
|         :Nick.WGDMX        :part.AP2WGDMX APLIMAGE
|         :Nick.WGPVS        :part.AP2WGPVS APLIMAGE
|         :Nick.WGRAP        :part.AP2WGRAP APLIMAGE
|         :Nick.WHOST        :part.AP2WHOST APLIMAGE
|         :Nick.WMATH        :part.AP2WMATH APLIMAGE
|         :Nick.WMEDI        :part.AP2WMEDI APLIMAGE
|         :Nick.WMESS        :part.AP2WMESS APLIMAGE
|         :Nick.WPRIN        :part.AP2WPRIN APLIMAGE
|         :Nick.WSQL         :part.AP2WSQL  APLIMAGE
|         :Nick.WSUPP        :part.AP2WSUPP APLIMAGE
|         :Nick.WTRAN        :part.AP2WTRAN APLIMAGE
|         :Nick.WUTIL        :part.AP2WUTIL APLIMAGE
|         :Nick.WVAPL        :part.AP2WVAPL APLIMAGE
|         :Nick.WVSAM        :part.AP2WVSAM APLIMAGE
|         :Nick.WWSIN        :part.AP2WWSIN APLIMAGE
|  :Nick.File7        :msg.FILE 7 - MACROS, TEXT DECKS
|                     :list. F7-MACROS TEXT-list
|    :Nick.F7-MACROS  :list. VOPT TAG RESAP VPTV PTH XPTX XVCT CHECK
|                            ECV CSVPE PCV SCV GSPC FSPC SDF CDR
|       :Nick.VOPT         :part.AP2VOPT  COPY
|       :Nick.TAG          :part.AP2TAG   COPY
|       :Nick.RESAP        :part.AP2RESAP COPY
|       :Nick.VPTV         :part.AP2VPTV  COPY
|       :Nick.PTH          :part.AP2PTH   COPY
|       :Nick.XPTX         :part.AP2XPTX  COPY
|       :Nick.XVCT         :part.AP2XVCT  COPY
|       :Nick.CHECK        :part.AP2CHECK COPY
|       :Nick.ECV          :part.AP2ECV   COPY
|       :Nick.GSPC         :part.AP2GSPC  COPY
|       :Nick.FSPC         :part.AP2FSPC  COPY
|       :Nick.CSVPE        :part.AP2CSVPE COPY
|       :Nick.PCV          :part.AP2PCV   COPY
|       :Nick.SCV          :part.AP2SCV   COPY
|       :Nick.SDF          :part.AP2SDF   COPY
|       :Nick.CDR          :part.AP2CDR   COPY
|  :Nick.File8        :msg.FILE 8 - MIGRATION AIDS FOR VS APL APs
|                     :list. F8-MV1AP F8-VASVP F8-VSMAC F8-MACROS
|       :Nick.F8-MV1AP   :part.AP2MV1AP EXEC
```

```
|       :Nick.F8-VASVP   :part.AP2VASVP
|       :Nick.F8-VSMAC   :part.AP2VSAPL MACLIB
|    :Nick.F8-MACROS  :list. F8M-VS F8M-SV F8M-A2
|      :Nick.F8M-VS   :list. APLDESC  APLKSTOZ APLKZTOS APLPCV   APLREGS
|                            APLSCV   APLSHSVP APLZCODE
|      :Nick.F8M-SV   :list. ASVDFORM ASVPACC  ASVPCPY  ASVPOFR  ASVPQRY
|                            ASVPREF ASVPRET   ASVPSOF  ASVPSON  ASVPSPC
|                            ASVPWAIT
|      :Nick.F8M-A2   :list. SDF CDR
|       :Nick.APLDESC      :part.APLDESC  COPY
|       :Nick.APLKSTOZ     :part.APLKSTOZ COPY
|       :Nick.APLKZTOS     :part.APLKZTOS COPY
|       :Nick.APLPCV       :part.APLPCV   COPY
|       :Nick.APLREGS      :part.APLREGS  COPY
|       :Nick.APLSCV       :part.APLSCV   COPY
|       :Nick.APLSHSVP     :part.APLSHSVP COPY
|       :Nick.APLZCODE     :part.APLZCODE COPY
|       :Nick.ASVDFORM     :part.ASVDFORM COPY
|       :Nick.ASVPACC      :part.ASVPACC  COPY
|       :Nick.ASVPCPY      :part.ASVPCPY  COPY
|       :Nick.ASVPOFR      :part.ASVPOFR  COPY
|       :Nick.ASVPQRY      :part.ASVPQRY  COPY
|       :Nick.ASVPREF      :part.ASVPREF  COPY
|       :Nick.ASVPRET      :part.ASVPRET  COPY
|       :Nick.ASVPSOF      :part.ASVPSOF  COPY
|       :Nick.ASVPSON      :part.ASVPSON  COPY
|       :Nick.ASVPSPC      :part.ASVPSPC  COPY
|       :Nick.ASVPWAIT     :part.ASVPWAIT COPY

|  :Nick.File9        :msg.FILE 9 - 3800 EXECS AND FONTS
|                     :list. FONT38  FONTPP  FONTnn

|    :Nick.FONT38     :rem. EXECs and fonts for 3800 line printers
|                     :list. G3800-x F3800-c G3803-x F3803-c M3800
|                            FONTS1 FONTS2 FONTS3 FONTS4

|       :Nick.G3800-x      :part.AP2G3800 EXEC
|       :Nick.F3800-c      :part.AP2F3800 CNTRL
|       :Nick.G3803-x      :part.AP2G3803 EXEC
|       :Nick.F3803-c      :part.AP2F3803 CNTRL
|       :Nick.M3800        :part.AP2M3800 EXEC
|    :Nick.FONTS1     :list. D0G1 D0XT D2G1 D2XT G0G1 G0XT G2G1 G2XT
|    :Nick.FONTS2     :list. G5G1 G5XT I0G1 I0XT I2G1 I2XT PLG1 PLXT
|    :Nick.FONTS3     :list. T0G1 T0XT T2G1 T2XT D0G3 D2G3 G0G3 G2G3
|    :Nick.FONTS4     :list. G5G3 I0G3 I2G3 PLG3 T0G3 T2G3

|       :Nick.D0G1         :part.AP2AD0G1 FONT
|       :Nick.D0XT         :part.AP2AD0XT FONT
|       :Nick.D2G1         :part.AP2AD2G1 FONT
|       :Nick.D2XT         :part.AP2AD2XT FONT
|       :Nick.G0G1         :part.AP2AG0G1 FONT
|       :Nick.G0XT         :part.AP2AG0XT FONT
|       :Nick.G2G1         :part.AP2AG2G1 FONT
|       :Nick.G2XT         :part.AP2AG2XT FONT
```

```
|       :Nick.G5G1         :part.AP2AG5G1 FONT
|       :Nick.G5XT         :part.AP2AG5XT FONT
|       :Nick.I0G1         :part.AP2AI0G1 FONT
|       :Nick.I0XT         :part.AP2AI0XT FONT
|       :Nick.I2G1         :part.AP2AI2G1 FONT
|       :Nick.I2XT         :part.AP2AI2XT FONT
|       :Nick.PLG1         :part.AP2APLG1 FONT
|       :Nick.PLXT         :part.AP2APLXT FONT
|       :Nick.T0G1         :part.AP2AT0G1 FONT
|       :Nick.T0XT         :part.AP2AT0XT FONT
|       :Nick.T2G1         :part.AP2AT2G1 FONT
|       :Nick.T2XT         :part.AP2AT2XT FONT
|       :Nick.D0G3         :part.AP2AD0G3 FONT
|       :Nick.D2G3         :part.AP2AD2G3 FONT
|       :Nick.G0G3         :part.AP2AG0G3 FONT
|       :Nick.G2G3         :part.AP2AG2G3 FONT
|       :Nick.G5G3         :part.AP2AG5G3 FONT
|       :Nick.I0G3         :part.AP2AI0G3 FONT
|       :Nick.I2G3         :part.AP2AI2G3 FONT
|       :Nick.PLG3         :part.AP2APLG3 FONT
|       :Nick.T0G3         :part.AP2AT0G3 FONT
|       :Nick.T2G3         :part.AP2AT2G3 FONT

|    :Nick.FONTnn     :rem. Fonts for 3812 and 3820 page printers
|                     :list. FONT38nn CSET3812 CSET3820

|    :rem. Code Page and Coded Fonts for 3812 and 3820 page printers
|    :rem. The typeface name is 'APL2 DOCUMENT FONT'
|    :rem. These fonts are all medium italic, normal width
|      :Nick.FONT38nn  :list. X06B X07B X08B X09B X00B X0AB
|                             X0BB X0DB X0FB X0HB X0JB X0NB T0293
|        :Nick.T0293   :part.T1200293 FONT3820  :rem. Code page 293
|        :Nick.X06B    :part.X081556B FONT3820  :rem.  6 point APL font
|        :Nick.X07B    :part.X081557B FONT3820  :rem.  7 point APL font
|        :Nick.X08B    :part.X081558B FONT3820  :rem.  8 point APL font
|        :Nick.X09B    :part.X081559B FONT3820  :rem.  9 point APL font
|        :Nick.X00B    :part.X081550B FONT3820  :rem. 10 point APL font
|        :Nick.X0AB    :part.X08155AB FONT3820  :rem. 11 point APL font
|        :Nick.X0BB    :part.X08155BB FONT3820  :rem. 12 point APL font
|        :Nick.X0DB    :part.X08155DB FONT3820  :rem. 14 point APL font
|        :Nick.X0FB    :part.X08155FB FONT3820  :rem. 16 point APL font
|        :Nick.X0HB    :part.X08155HB FONT3820  :rem. 18 point APL font
|        :Nick.X0JB    :part.X08155JB FONT3820  :rem. 20 point APL font
|        :Nick.X0NB    :part.X08155NB FONT3820  :rem. 24 point APL font

|      :Nick.CSET3812  :rem. Character Sets for 3812 page printers
|                      :list. C260 C270 C280 C290 C200 C2A0
|                             C2B0 C2D0 C2F0 C2H0 C2J0 C2N0
|        :Nick.C260    :part.C0815560 FONT3812  :rem.  6 point char defs
|        :Nick.C270    :part.C0815570 FONT3812  :rem.  7 point char defs
|        :Nick.C280    :part.C0815580 FONT3812  :rem.  8 point char defs
|        :Nick.C290    :part.C0815590 FONT3812  :rem.  9 point char defs
|        :Nick.C200    :part.C0815500 FONT3812  :rem. 10 point char defs
|        :Nick.C2A0    :part.C08155A0 FONT3812  :rem. 11 point char defs
```

```
|        :Nick.C2B0    :part.C08155B0 FONT3812  :rem. 12 point char defs
|        :Nick.C2D0    :part.C08155D0 FONT3812  :rem. 14 point char defs
|        :Nick.C2F0    :part.C08155F0 FONT3812  :rem. 16 point char defs
|        :Nick.C2H0    :part.C08155H0 FONT3812  :rem. 18 point char defs
|        :Nick.C2J0    :part.C08155J0 FONT3812  :rem. 20 point char defs
|        :Nick.C2N0    :part.C08155N0 FONT3812  :rem. 24 point char defs

|      :Nick.CSET3820  :rem. Character Sets for 3820 page printers
|                      :list. C060 C070 C080 C090 C000 C0A0
|                             C0B0 C0D0 C0F0 C0H0 C0J0 C0N0
|        :Nick.C060    :part.C0815560 FONT3820  :rem.  6 point char defs
|        :Nick.C070    :part.C0815570 FONT3820  :rem.  7 point char defs
|        :Nick.C080    :part.C0815580 FONT3820  :rem.  8 point char defs
|        :Nick.C090    :part.C0815590 FONT3820  :rem.  9 point char defs
|        :Nick.C000    :part.C0815500 FONT3820  :rem. 10 point char defs
|        :Nick.C0A0    :part.C08155A0 FONT3820  :rem. 11 point char defs
|        :Nick.C0B0    :part.C08155B0 FONT3820  :rem. 12 point char defs
|        :Nick.C0D0    :part.C08155D0 FONT3820  :rem. 14 point char defs
|        :Nick.C0F0    :part.C08155F0 FONT3820  :rem. 16 point char defs
|        :Nick.C0H0    :part.C08155H0 FONT3820  :rem. 18 point char defs
|        :Nick.C0J0    :part.C08155J0 FONT3820  :rem. 20 point char defs
|        :Nick.C0N0    :part.C08155N0 FONT3820  :rem. 24 point char defs

|    :Nick.FONTPP     :rem. Fonts for 3800-3 page printers
|                     :list. FONT38PP CSET38PP

|    :rem. Code Page and Coded Fonts for 3800 page printers
|    :rem. The typeface name is 'APL2 DOCUMENT FONT'
|    :rem. These fonts are all medium italic, normal width
|      :Nick.FONT38PP  :list. X16B X17B X18B X19B X10B X1AB
|                             X1BB X1DB X1FB X1HB X1JB X1NB T1293
|        :Nick.T1293   :part.T1200293 FONT38PP  :rem. Code page 293
|        :Nick.X16B    :part.X181556B FONT38PP  :rem.  6 point APL font
|        :Nick.X17B    :part.X181557B FONT38PP  :rem.  7 point APL font
|        :Nick.X18B    :part.X181558B FONT38PP  :rem.  8 point APL font
|        :Nick.X19B    :part.X181559B FONT38PP  :rem.  9 point APL font
|        :Nick.X10B    :part.X181550B FONT38PP  :rem. 10 point APL font
|        :Nick.X1AB    :part.X18155AB FONT38PP  :rem. 11 point APL font
|        :Nick.X1BB    :part.X18155BB FONT38PP  :rem. 12 point APL font
|        :Nick.X1DB    :part.X18155DB FONT38PP  :rem. 14 point APL font
|        :Nick.X1FB    :part.X18155FB FONT38PP  :rem. 16 point APL font
|        :Nick.X1HB    :part.X18155HB FONT38PP  :rem. 18 point APL font
|        :Nick.X1JB    :part.X18155JB FONT38PP  :rem. 20 point APL font
|        :Nick.X1NB    :part.X18155NB FONT38PP  :rem. 24 point APL font

|      :Nick.CSET38PP  :rem. Character Sets for 3800 page printers
|                      :list. C160 C170 C180 C190 C100 C1A0
|                             C1B0 C1D0 C1F0 C1H0 C1J0 C1N0
|        :Nick.C160    :part.C1815560 FONT38PP  :rem.  6 point char defs
|        :Nick.C170    :part.C1815570 FONT38PP  :rem.  7 point char defs
|        :Nick.C180    :part.C1815580 FONT38PP  :rem.  8 point char defs
|        :Nick.C190    :part.C1815590 FONT38PP  :rem.  9 point char defs
|        :Nick.C100    :part.C1815500 FONT38PP  :rem. 10 point char defs
|        :Nick.C1A0    :part.C18155A0 FONT38PP  :rem. 11 point char defs
```

```
|       :Nick.C1B0    :part.C18155B0 FONT38PP  :rem. 12 point char defs
|       :Nick.C1D0    :part.C18155D0 FONT38PP  :rem. 14 point char defs
|       :Nick.C1F0    :part.C18155F0 FONT38PP  :rem. 16 point char defs
|       :Nick.C1H0    :part.C18155H0 FONT38PP  :rem. 18 point char defs
|       :Nick.C1J0    :part.C18155J0 FONT38PP  :rem. 20 point char defs
|       :Nick.C1N0    :part.C18155N0 FONT38PP  :rem. 24 point char defs

| :Nick.t-WTM      :cmd.VMFPLC2 WTM

| :rem.*---------------------------------------------------------------*
| *
| *  I5688229 Tags - APL2AE Tape Definition
| *
| *---------------------------------------------------------------*

| :Nick.AETAPE       :msg.*** SYSTEM APL2AE &-LV
|                    :list.AEFile1 t-WTM AEFile2 t-WTM AEFile3 t-WTM
|                          AEFile4 t-WTM AEFile5 t-WTM AEFile6 t-WTM
|                          AEFile7 t-WTM t-WTM

| :Nick.AEFile1      :msg.FILE 1 - INSTALL LEVEL and EXEC
|                    :list.AEF1-INSTLVL  AEF1-INSTEXEC
|      :Nick.AEF1-INSTLVL   :part.I5688229 022007
|      :Nick.AEF1-INSTEXEC  :part.I5688229 EXEC
| :Nick.AEFile2      :msg.FILE 2 - MEMO TO USERS
|                    :list.AEF2-MEMO
|      :Nick.AEF2-MEMO      :part.I5688229 MEMO
| :Nick.AEFile3      :msg.FILE 3 - INSTALLATION, SERVICE, and SAMPLE FILES
|                    :list.AEF3-service AEF3-samples AEF3-nrs
|                          F3-lang F3-help F3-prof
|   :Nick.AEF3-service :list.F3-VIOPE F3-VIAPL F3-M220 F3-MAINT F3-MLNK
|                           F3-MCOPY F3-MV2AP F3-MP11L F3-MP11M F3-MREN
|                           F3-VMCF  F3-VUTIL F3-XLOAD F3-APL2PI
|                           F3-CSVP F3-C220 F3-E220-m F3-E220-l F3-F220
|      :Nick.F3-APL2PI    :part.APL2PI   MODULE
|      :Nick.F3-CSVP      :part.AP2CSVP  MODULE
|      :Nick.F3-C220      :part.AP2C220  LOADLIB
|      :Nick.F3-E220-l    :part.AP2E220  LOADLIB
|      :Nick.F3-E220-m    :part.AP2E220  MODULE
|      :Nick.F3-F220      :part.AP2F220  LOADLIB
|      :Nick.F3-MAINT     :part.AP2MAINT EXEC
|      :Nick.F3-MCOPY     :part.AP2MCOPY EXEC
|      :Nick.F3-MLNK      :part.AP2MLNK  EXEC
|      :Nick.F3-MP11L     :part.AP2MP11L EXEC
|      :Nick.F3-MP11M     :part.AP2MP11M EXEC
|      :Nick.F3-MREN      :part.AP2MREN  EXEC
|      :Nick.F3-MV2AP     :part.AP2MV2AP EXEC
|      :Nick.F3-M220      :part.AP2M220  NAMES
|      :Nick.F3-VIAPL     :part.AP2VIAPL
|      :Nick.F3-VIOPE     :part.AP2VIOPE
|      :Nick.F3-VMCF      :part.AP2VMCF  MODULE
|      :Nick.F3-VUTIL     :part.AP2VUTIL MODULE
|      :Nick.F3-XLOAD     :part.AP2XLOAD MODULE
|   :Nick.AEF3-samples :list.F3-AP2EXIT  F3-AP2MPSRV F3-AP2LOCAL F3-MACLIB
```

```
|                            F3-NAME3800 F3-RDEVICE  F3-NAMESYS  F3-LIBTAB
|                            F3-AP2SVPEX F3-XAPICSAM F3-VIOPESAM F3-SQLI

|       :Nick.F3-AP2EXIT   :part.AP2EXIT  AP2MSAMP
|       :Nick.F3-AP2LOCAL  :part.AP2LOCAL AP2MSAMP
|       :Nick.F3-AP2MPSRV  :part.AP2MPSRV AP2MSAMP
|       :Nick.F3-AP2SVPEX  :part.AP2SVPEX AP2MSAMP
|       :Nick.F3-MACLIB    :part.AP2MAC   MACLIB
|       :Nick.F3-NAMESYS   :part.NAMESYS  AP2MSAMP
|       :Nick.F3-NAME3800  :part.NAME3800 AP2MSAMP
|       :Nick.F3-RDEVICE   :part.RDEVICE  AP2MSAMP
|       :Nick.F3-SQLI      :part.AP2V127I ASMSQL
|       :Nick.F3-VIOPESAM  :part.AP2VIOPE AP2MSAMP
|       :Nick.F3-XAPICSAM  :part.AP2XAPIC AP2MSAMP
|    :Nick.AEF3-nrs     :list.F3-VNRP1 F3-VNRP2 F3-VNRP3 F3-VNRP4
|                            F3-VNRP5 F3-VNRPS
|       :Nick.F3-VNRP1     :part.AP2VNRP1 LOADLIB
|       :Nick.F3-VNRP2     :part.AP2VNRP2 LOADLIB
|       :Nick.F3-VNRP3     :part.AP2VNRP3 LOADLIB
|       :Nick.F3-VNRP4     :part.AP2VNRP4 LOADLIB
|       :Nick.F3-VNRP5     :part.AP2VNRP5 LOADLIB
|       :Nick.F3-VNRPS     :part.AP2VNRPS LOADLIB
|  :Nick.AEFile4      :msg.FILE 4 - ONLINE HELP
|                     :list.F4-HELP-APL2AE
|  :Nick.AEFile5      :msg.FILE 5 - RUN-TIME AUXILIARY FILES
|                     :list.F5-V5688229 F5-AP2VN011
|                           F5-usertext
|  :Nick.AEFile6      :msg.FILE 6 - MACROS, TEXT DECKS
|                     :list. F7-MACROS AETEXT-list
|  :Nick.AEFile7      :msg.FILE 7 - 3800 EXECS AND FONTS
|                     :list. FONT38  FONTPP  FONTnn
```

Appendix G. SCRIPT/VS Support

This appendix describes the modifications to be made to SCRIPT/VS Release 3 in order for the SCRIPT/VS user to exploit the APL2 fonts. A complete description of the modification procedure is contained in *Document Composition Facility SCRIPT/VS Language Reference*.

Box Character Set

Code the DSMBCS macro to define the APL2 box characters as follows:

```
DSMBCS NAME=APL,                                                00010000
       CROSS=2C,                                                00020000
       VBAR=1A,                                                 00030000
       HBAR=2D,                                                 00040000
       ULCORNR=1C,                                              00050000
       URCORNR=1B,                                              00060000
       LLCORNR=1E,                                              00070000
       LRCORNR=1F,                                              00080000
       TOPTEE=3B,                                               00090000
       BOTTEE=3E,                                               00100000
       LEFTEE=3D,                                               00110000
       RIGHTEE=3F                                               00120000
```

Line Printer Font Information

Code the DSMFIB macros to define the APL2 fonts as follows:

```
DSMFIB   NAME=AD10,WTABLE=PROP10,BCS=APL,PITCH=10,FS=4           00010000
DSMFIB   NAME=AD12,WTABLE=PROP12,BCS=APL,PITCH=12,FS=4           00020000
DSMFIB   NAME=AG10,WTABLE=PROP10,BCS=APL,PITCH=10,FS=4           00030000
DSMFIB   NAME=AG12,WTABLE=PROP12,BCS=APL,PITCH=12,FS=4           00040000
DSMFIB   NAME=AG15,WTABLE=PROP15,BCS=APL,PITCH=15,FS=4           00050000
DSMFIB   NAME=AI10,WTABLE=MONO10,BCS=VAN,PITCH=10,FS=2           00060000
DSMFIB   NAME=AI12,WTABLE=MONO12,BCS=VAN,PITCH=12,FS=2           00070000
DSMFIB   NAME=APL,WTABLE=MONO10,BCS=APL,PITCH=10,FS=4            00080000
DSMFIB   NAME=AT10,WTABLE=PROP10,BCS=APL,PITCH=10,FS=4           00090000
DSMFIB   NAME=AT12,WTABLE=PROP12,BCS=APL,PITCH=12,FS=4           00100000
```


© Copyright IBM Corp. 1984, 1994

Bibliography

APL2 Publications

- *APL2 Fact Sheet,* GH21-1090
- *APL2/370 Application Environment Licensed Program Specifications,* GH21-1063
- *APL2/370 Licensed Program Specifications,* GH21-1070
- *APL2 for AIX/6000 Licensed Program Specifications,* GC23-3058
- *APL2 for Sun Solaris Licensed Program Specifications,* GC26-3359
- *APL2/370 Installation and Customization under CMS,* SH21-1062
- *APL2/370 Installation and Customization under TSO,* SH21-1055
- *APL2 Migration Guide,* SH21-1069
- *APL2 Programming: Language Reference,* SH21-1061
- *APL2/370 Programming: Processor Interface Reference,* SH21-1058
- *APL2 Reference Summary,* SX26-3999
- *APL2 Programming: An Introduction to APL2,* SH21-1073
- *APL2 for AIX/6000: User's Guide,* SC23-3051
- *APL2 for OS/2: User's Guide,* SH21-1091
- *APL2 for Sun Solaris: User's Guide,* SH21-1092
- *APL2 for the IBM PC: User's Guide,* SC33-0600
- *APL2 GRAPHPAK: User's Guide and Reference,* SH21-1074
- *APL2 Programming: Using Structured Query Language,* SH21-1057
- *APL2/370 Programming: Using the Supplied Routines,* SH21-1056
- *APL2/370 Programming: System Services Reference,* SH21-1054
- *APL2/370 Diagnosis Guide,* LY27-9601
- *APL2/370 Messages and Codes,* SH21-1059

Other Books You Might Need

The following book is recommended:

- *APL2 at a Glance*, by James Brown, Sandra Pakin, and Raymond Polivka, published by Prentice-Hall, ISBN 0-13-038670-7 (1988). (Copies can be ordered from IBM as SC26-4676.)

Plastic replacement keyboard keycaps are included with this product. Additional sets of keyboard keycaps are available from IBM as:

- *APL2 Keycaps (US and UK base set)*, SX80-0270
- *APL2 Keycaps, German upgrade to SX80-0270*, SX23-0452
- *APL2 Keycaps, French upgrade to SX80-0270*, SX23-0453
- *APL2 Keycaps, Italian upgrade to SX80-0270*, SX23-0454.

Two sets of *APL2 Keyboard Decals*, SC33-0604, are included with this product. Additional sets of these decal sheets can be ordered from IBM.

Assembler Language

If you are customizing (updating and assembling source code):

- *OS/VS, DOS/VS, and VM/370 Assembler Language*, GC33-4010

GDDM

If you are using the Graphical Data Display Manager (GDDM):

- *GDDM General Information*, GC33-0866
- *GDDM Installation and System Management for VM*, GC33-0873

IBM 3800 Printer Fonts

- *IBM 3800 Printing Subsystem Programmer's Guide*, GC26-3846
- *IBM 3800 Printing Subsystem Model 3 Programmer's Guide: Compatibility*, SH35-0051

SQL/DS

- *SQL/Data System Messages and Codes*, SH09-8079
- *SQL/Data System Application Programmer's Guide*, SH09-8086

© Copyright IBM Corp. 1984, 1994

VM/XA

- *VM/XA System Product Installation and Service*, SC23-0364
- *VM/XA System Product Planning and Administration*, GC23-0378
- *VM/XA System Product CP Command Reference*, SC23-0358
- *VM/XA System Product CMS Command Reference*, SC23-0354
- *VM/XA System Product Library Guide, Glossary, and Master Index*, GC23-0367
- *VM/XA System Messages and Codes Reference*, SC23-0376

VM/ESA

The following are VM/ESA Common books:

- *VM/ESA CMS Command Reference*, SC24-5461

The following are VM/ESA (ESA Feature) books:

- *VM/ESA Installation*, SC24-5526
- *VM/ESA CP Planning and Administration*, SC24-5521
- *VM/ESA Library Guide and Master Index*, GC24-5518
- *VM/ESA CP Command and Utility Reference*, SC24-5519
- *VM/ESA System Messages and Codes Reference*, SC24-5529

The following are VM/ESA (370 Feature) books:

- *VM/ESA Installation for 370*, SC24-5431
- *VM/ESA CP Planning and Administration for 370*, SC24-5430
- *VM/ESA CP General User Command Reference for 370*, SC24-5433
- *VM/ESA Library Guide and Master Index for 370*, GC24-5436
- *VM/ESA System Messages and Codes for 370*, SC24-5437

For corequisite publications for other VM systems, see the appropriate *VM Library Guide and Master Index*.

VS APL

If you are adding user-written VS APL style auxiliary processors to your system:

- *VS APL for CMS and TSO: Writing Auxiliary Processors*, SH20-9068

Index

Special Characters

)*MORE* system command 34
□*AI* account information, first item of 37

Numerics

3277 model 2 35
3278 model 2 35
3800 font support 48, 51, 52

A

A-disk space requirements 82
adding saved segment information 52
ADMK772A device-token 35
ADMK782A device-token 35
□*AI* account information, first item of 37
altering normal APL2 debugging actions (DEBUG invocation option) 33
AP2EXIT EXEC
  modifying 64
  sample 83, 84
AP2EXIT EXEC (CMS), ending your APL2 session 44
AP2LOCAL NAMES
  customizing APL2 47
  sample 85, 90
AP2M EXECs 47
AP2M220 NAMES 47, 135, 167
AP2M3800 EXEC 48
AP2MAINT EXEC 47
AP2MCOPY EXEC 48
AP2MLNK EXEC 47
AP2MP11L EXEC 48
AP2MP11M EXEC 48
AP2MREN EXEC 48
AP2MV2AP EXEC 48
AP2MWS EXEC 48
AP2SVPEX, sample exit for global SVP 92
AP2VIOPE ASSEMBLE sample 104
AP2VIOPT ASSEMBLE sample 107
AP2VOPT macro 129
APL2
  A-disk space requirements 82
  customizing 46
  customizing execution 61
  data security 13
  data sets 133
  environment, administering the 74
  files 133
  GDDM 13
  installation planning 3
  installation process, customizing 46
APL2 *(continued)*
  installation process, overview 2
  installation review 21
  installation storage requirements 11
  invocation 25
  invocation options 131, 132
  libraries 5, 64
  maintaining 78, 80
  migration from previous APL2 releases 12
  migration from Release 1 12
  migration from VS APL 11, 12
  modifying or adding national language help files 71
  modules 5
  national language message and command files 70
  overview 1, 2
  packaging in CMS 1, 2
  port server user ID, sample 82
  post-installation customizing requirements 13
  problem documentation 78
  problems 78
  running global SVP service machine 74, 75
  sample installation 120, 124
  session manager profile 61
  SQL/DS access 68
  storage requirements 11
  system files 5
  system options 129, 130
  termination 25
  user user ID, sample 81
APL2 command
  overriding default invocation options 26
APLSCOPT TEXT deck 53
APNAMES invocation option 28
Application Environment
  facilities not provided with 1
  highlights 1
associated processor 11 55
auxiliary processors
  module names of 30
  supplied with APL2 30
  user-written
    creating 53
    making resident 54
    testing 53

B

box character set 168

© Copyright IBM Corp. 1984, 1994

C

CASE 27, 30
changes
  summary of x
CMS
  DD names 134
  defining APL2 libraries 64, 67
  disk tables for CMS devices 82
  filedef names 134
  installation exit for global SVP 67, 68
  installation summary 14
  packaging of APL2 1, 2
CODE invocation option 31
continuing your terminal session after exiting APL2
  CMS 44
copying files to production disks 21, 23, 24
corrective service, maintaining APL2 79
customizing APL2
  execution
    accessing SQL/DS 68, 69
    AP2EXIT EXEC, modifying 64
    defining libraries 64, 67
    installation exit for global SVP 67, 68
    modifying or adding national language help files 71
    nondefault saved segment names for APL2 63
    session manager profile 61
    system options 62, 63
  extending or modifying national language message and command files 70
  IBM-supplied sample files 46
  installation process
    3800 font support 51, 52
    adding saved segment information 52
    AP2M EXECs 47
    customizing saved segment information 52
    installing to different disk 49, 51
    modified APLSCOPT TEXT deck 53
    NAMES files 46, 47
    NAMES files tags 48, 49
    performing 16
    performing customized installation 17
    removing GDDM 51
    removing SQL/DS 51
    retrieving optional material 49, 51
    steps 49
    testing APL2 without saving segments 52
    user-written auxiliary processors 53, 55
    user-written external routines 55
  post-installation requirements 13
  system options 17
customizing saved segment information 52

D

data security, planning for 13
data sets, APL2 133
date and time stamp display 32
DATEFORM invocation option 32
DBCS conversion option 32
DBCS invocation option 32
DCSS (Discontiguous Shared Segments)
  considerations for installation 8, 10
  defining a shared code 9
  defining a shared code saved segment 8
  defining global shared storage 10
  nondefault names for APL2 63
  overview 1
DD names 134
DEBUG invocation option 33
debugging APL2
  DEBUG invocation option 33
  SYSDEBUG invocation option 42
  TRACE invocation option 43
default installation 17
default invocation options 131
default library 66
default library number (first item of $\square AI$) 37
DEFAULT VSAPLPR sample 113
device-token, DSOPEN invocation option 35
diagnosing problems 42, 78
disconnection of APL2
  CMS 45
disk tables, CMS devices 82
display of output, preventing
  QUIET invocation option 39
display, control of the screen, using AP 126 39
documenting problems 78
DSOPEN invocation option 35

E

ending your APL2 session
  CMS 44
EXCLUDE invocation option 35
execution, customizing
  accessing SQL/DS 68, 69
  AP2EXIT EXEC, modifying 64
  APL2 61
  defining libraries
    default library 66
    library number 64, 65
    with LIBTAB APL2 66, 67
  installation exit for global SVP 67, 68
  nondefault saved segment names for APL2 63
  session manager profile 61
  system options 62, 63
external routines
  link-editing
    with APL2 60

external routines *(continued)*
packaged for APL2 55
user-written 55

F

filedef (DD) names 134
files, APL2 133
font support, 3800 48, 51, 52
forced termination of APL2
CMS 45
FREESIZE invocation option 36

G

GDDM (Graphical Data Display Manager)
access 18
DSOPEN function 35
FREESIZE function 36
in customizing APL2 51
installation planning 3
overview 2
removing 51
GDDM, planning for 13
global shared storage saved segment 10
in AP2VIOPE 10
global Shared Variable Processor (SVP)
considerations for installation 6, 7
controlling its operation 74, 75
copying during installation 22
installation exit 67, 68
installation planning 6
migration from Release 1 12
overview 2
running the service machine 74
sample exit 92, 100
user ID, sample 81, 82

H

hardware requirements 4
highlighting input/output 36
HILIGHT invocation option 36

I

ID invocation option 37
IMAGELIBs, defining 14
INPUT invocation option 38
INPUT 38
relationship to QUIET invocation option 39
installation
customized 16
customizing 17, 46, 55
EXEC 14, 15
EXEC return codes 20
procedure 14
installation *(continued)*
process, overview 2
review 21
sample
APL2 120
Application Environment 125
steps 14
summary 14
tags 48
user ID, defining 4
verification (IVP) 23, 24
installing APL2
copying files to production disks 21, 23, 24
customized installation, performing 16, 17
customizing 46, 55
customizing steps, outline 49
data security 13
default installation 17
defining a shared code saved segment 8, 9
defining global shared storage saved segment 10
GDDM 13
global SVP 6, 7
global SVP, installation exit 67
hardware requirements 4
installation EXEC and MEMO, loading 15, 16
installation review 21
library considerations 5
migration from previous APL2 releases 12
migration from Release 1 12
migration from VS APL 11, 12
minidisk, defining 4, 5
modules 5
overview 2
planning 3
post-installation customizing requirements 13
preplanning activity 3
procedure 14
product user IDs, defining 15
review of 21
sample 120, 124, 125, 128
saved segments 8, 10
saved segments, defining 14, 15
software requirements 3, 4
SQL/DS interface module 18
steps 14
storage requirements 11
summary 14
system files 5
system options in AP2VIOPE 10
system options in AP2VIOPT 10
system options, customizing 17, 62
system requirements 3
user ID, defining 4, 5
user ID, sample 81
interruption of APL2
CMS 45

invocation options 131, 132
APNAMES 131
CASE 131
changing 62, 63
DATEFORM 131
DBCS 131
DEBUG 131
default 131
DSOPEN 131
EXCLUDE 131
FREESIZE 131
HILIGHT 131
ID 132
INPUT 132
NLT 132
PROFILE 132
QUIET 132
RUN 132
SHRSIZE 132
SMAPL 132
SVMAX 132
SYSDEBUG 132
TERMCODE 42, 132
TRACE 132
WSSIZE 132
XA 132
invoking APL2 25

L

language message and command keyword definition files 70
libraries
considerations for installation 5
copying public libraries during installation 22
defining
default library 66
library number 64, 65
with LIBTAB APL2 66, 67
private 65
public 65
LIBTAB APL2
defining libraries 66, 67
sample 114, 115
line printer font information 168

M

maintaining APL2
corrective service 79, 80
overview 78
preventive service with PUT 78
memo to users 15, 16
message number
See MORE
message, providing a local greeting 72
migration
from previous APL2 releases 12
from Release 1 12
from VS APL 11, 12
minidisk, defining 4
modules 5
)*MORE* system command 34

N

NAME3800 macro sample 118
NAMES files
AP2LOCAL NAMES 47
AP2M220 NAMES 47, 135
tags 48, 49
NAMESYS macro sample 116, 117
NLT (National Language Tables), extending or modifying 70
NLT invocation option 38
number, APL2 library 64

O

optional material, retrieving 49
output
preventing
QUIET invocation option 39
overriding normal APL2 debugging actions 33

P

packaging in CMS 1, 2
planning the installation
data security 13
defining a shared code saved segment 8, 9
defining global shared storage saved segment 10
GDDM 13
global SVP 7
global SVP installation 6
hardware requirements 4
library considerations 5
migration from previous APL2 releases 12
migration from Release 1 12
migration from VS APL 11, 12
minidisk, defining 4, 5
modules 5
overview 3
post-installation customizing requirements 13
preplanning activity 3
saved segments 8, 10
software requirements 3, 4
storage requirements 11
system files 5
system options in AP2VIOPE 10
system options in AP2VIOPT 10

planning the installation *(continued)*
system requirements 3
user ID, defining 4, 5
port server 7
defined 7
planning for installation of 7
running the 75
post-installation customizing requirements 13
preventing display of output
QUIET invocation option 39
preventive service, maintaining APL2 78
private libraries
defining 65
problem diagnosis techniques 78
problems, documentation of 78
processor 11 55
product user IDs, defining 15
PROFILE invocation option 38
profile-prefix 38
PTF (Program Temporary Fix) 2, 79
public libraries
copying during installation 22
defining 65

Q

QUIET invocation option 39
QUOTA system command
maximum number of shared variables 42
size of active workspace 43
size of shared storage 40
QUOTA system command
maximum number of shared variables 42
size of active workspace 43
size of shared storage 40

R

RDEVICE macro sample 119
reconnecting APL2 after disconnection 45
recovery, altering normal APL2 recovery actions 33
resuming an APL2 session
CMS 45
return codes issued by installation EXEC 20

S

sample files
AP2EXIT EXEC 83, 84
AP2LOCAL NAMES 85, 90
AP2MPSRV AP2MSAMP 91
AP2SVPEX AP2MSAMP 92
AP2VIOPE ASSEMBLE 104, 106
AP2VIOPT ASSEMBLE 107, 109
AP2XAPIC AP2MSAMP 110, 112
customizing 46
sample files *(continued)*
DEFAULT VSAPLPR 113
exit for GSVP 92, 100
IBM-supplied 46
LIBTAB APL2 114, 115
NAME3800 macro 118
NAMESYS macro 116, 117
RDEVICE macro 119
sample 101, 103
sample installation 120, 128
sample installation 125, 128
sample user IDs
APL2 port server user ID 82
APL2 user user ID 81
global SVP user ID 81
installation user ID 81
saved segments, defining 14, 15
screen display control 39
SCRIPT/VS support
box character set 168
line printer font information 168
service
corrective 79
preventive 78
session manager profile 61
shared code saved segment 8
SHRSIZE invocation option 40
size
of active workspace 30, 43, 44
of shared variable storage 40
SMAPL invocation option 41
software requirements 3, 4
space requirements, A-disk 82
SQL/DS (Structured Query Language/Data System)
accessing
database administrators 69
individual users 68
gaining access to 18
in customizing APL2 51
interface module 18
overview 2
removing 51
STAE 2
STAX 2
steps for installation 14
STIMER 2
storage requirements 11
summary of changes x
suppression of APL2 termination messages 39
SVMAX invocation option 42
SYSDEBUG invocation option 42
system files 5
system options 129, 130
AP2HSNM 129
AP2SHNM 129
ATASKS 129

system options *(continued)*
changing 62, 63
CSVPID 129
customizing 17
DEFAULT 129
EXITCMD 129
GDMPRTY 129
in AP2VIOPE 10
in AP2VIOPT 10
OVERIDE 129
QNLT 129
QTZDEC 130
QTZINT 130
RESAPS 130
RESEPS 130
specified by the AP2VOPT macro 129
system requirements for installation
hardware 4
software 3, 4

T

techniques for diagnosing problems 78
terminal type, identifying to APL2
DSOPEN invocation option 35
terminating an APL2 session
CMS 44
terminating APL2 25
testing APL2 without saving segments 52
time and date stamp (DATEFORM invocation option) 32
TRACE invocation option 43

U

user ID
installation 4
product 15
user-to-user shared variable communication, ID invocation option 37

V

verification procedure (IVP) 23
VM PUT 78
VMCF (Virtual Machine Communication Facility) 2
VS APL text deck 18

W

WSSIZE invocation option 30, 43

X

XA 27
XA invocation option 44

File Number: S370-34
Program Number: 5688-228
5688-229

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

The APL2 Library

GH21-1090 APL2 Family of Products (fact sheet)
SH21-1073 APL2 Programming: An Introduction to APL2
SH21-1061 APL2 Programming: Language Reference
SX26-3999 APL2 Reference Summary
SH21-1074 APL2 GRAPHPAK: User's Guide and Reference
SH21-1057 APL2 Programming: Using Structured Query Language
SH21-1069 APL2 Migration Guide
SC33-0600 APL2 for the IBM PC: User's Guide
SC33-0601 APL2 for the IBM PC: Reference Summary
SC33-0851 APL2 for the IBM PC: Reference Card
SH21-1091 APL2 for OS/2: User's Guide
GC23-3058 APL2 for AIX/6000 Licensed Program Specifications
SC23-3051 APL2 for AIX/6000: User's Guide
GC26-3359 APL2 for Sun Solaris Licensed Program Specifications
SH21-1092 APL2 for Sun Solaris: User's Guide
GH21-1063 APL2/370 Application Environment Licensed Program Specifications
GH21-1070 APL2/370 Licensed Program Specifications
SH21-1062 APL2/370 Installation and Customization under CMS
SH21-1055 APL2/370 Installation and Customization under TSO
SH21-1054 APL2/370 Programming: System Services Reference
SH21-1056 APL2/370 Programming: Using the Supplied Routines
SH21-1058 APL2/370 Programming: Processor Interface Reference
LY27-9601 APL2/370 Diagnosis Guide
SH21-1059 APL2/370 Messages and Codes

SH21-1062-01